RADIO TEST SET

PROGRAMMING MANUAL

2965A, 2966A, 2967 & 2968

© Marconi Instruments Ltd. 1997

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, or recorded by any information storage or retrieval system, without permission in writing by Marconi Instruments Ltd.

Printed in the UK

Manual part no. 46882-280M
Issue 6

4 August 1997

MP MASTER

MA31138

About this manual

This manual explains how to program the 2965A, 2966A, 2967 and 2968, and their cellular and trunking systems, for both automatic testing and remote control.

Intended audience

Persons engaged on work relating to the automatic testing of RF communications equipment. It is assumed that the reader will be familiar with telecommunication terms used in cellular and trunking radio systems.

Structure

Part 1 contains
Chapters 1, 2 and 3

These chapters cover all aspects of programming in MI-BASIC.

Part 2 contains
Chapters 4, 5 and 6

These chapters cover all aspects of programming to remotely control the 2965A, 2966A, 2967 or 2968 over the GPIB or RS232 interfaces.

Associated publications

Other manuals that cover specific aspects of this test set are:-

- **Operating Manual for 2965A, 2966A, 2967 and 2968** (46882-274T) provides overall test set information.
- **TETRA Supplement** (46882-324Z) provides testing details for TETRA systems.
- **GSM/DCS/PCS Supplement** (46882-165V) provides testing details for GSM/DCS/PCS systems.
- **NMT Supplement** (46882-275P) provides testing details for NMT systems.
- **AMPS Supplement** (46882-276X) provides testing details for AMPS systems.
- **TACS Supplement** (46882-277M) provides testing details for TACS systems.
- **MPT1327 Supplement** (46882-278C) provides testing details for MPT systems.
- **PMR Supplement** (46882-279R) provides testing details for PMR systems.

Contents

Tab number

PART 1 MI-BASIC

Chapter 1 GETTING STARTED 1

Chapter 2 MI-BASIC REFERENCE GUIDE 2

Chapter 3 TEST PARAMETERS 3

PART 2 REMOTE PROGRAMMING/CONTROL

Chapter 4 INTRODUCTION TO REMOTE CONTROL 4

Chapter 5 INSTRUMENT COMMANDS 5

Chapter 6 SYSTEMS COMMANDS 6

Chapter 1
GETTING STARTED

Contents

What is MI-BASIC? 1-1
MI-BASIC and the test set 1-1
Writing programs 1-1
Program example 1-2
Downloading programs; PC to test set 1-3
Step-by-step procedure for downloading a program 1-3
Storing and recalling programs on memory cards 1-6

List of figures

Fig. 1-1 System selection menu 1-4
Fig. 1-2 Program menu 1-4

What is MI-BASIC?

The MI-BASIC interpreter resides in the test set and runs either internal systems programs or user-defined programs that are written and downloaded to the test set via RS232 or GPIB.

MI-BASIC programs written by the user can set and measure either RF or AF frequencies on the N-Type, TNC and BNC connectors, send or receive information to or from a device over RS232 and set logic levels on four pins of the ACCESSORY socket.

Internal tests, specific to radio testing, can be included and executed by MI-BASIC programs. Some protocols are specific to Cellular or Trunking radios, the remainder are general parametric tests.

MI-BASIC programs can be run via the keyboard or in remote control via RS232 or GPIB. Text can be entered in upper, lower or mixed case.

MI-BASIC and the test set

MI-BASIC adds a controller facility to the test set and its facilities fall into several categories; program control, reading the keyboard, manipulation of variables, printing to either the display and/or results store, control of external hardware via the accessory port or RS232, parametric measurements, and the execution of internal tests.

Writing programs

The MI-BASIC statements are defined in Chapter 2 of this manual. An MI-BASIC program must be written in accordance with the statements and syntax described in that chapter.

A program can be written using any text editor. The recommended layout for a program is as follows:-

LABELS and **REM** statements should be at the left margin. The rest of the program should be one tab in from the left margin.

Where multiple-line **IF** statements are used, the program segment following the **THEN** part of the **IF** statement should be on the next line and one tab in from the start of the **IF** statement.

Subroutines should be placed after the **END** statement.

An example of a simple program is given on the following page. This program puts up a menu and requests that the user selects one of the options by using the data keys. When an option has been selected, the program branches to the appropriate subroutine. Once the subroutine has been executed, the program loops back to the menu and the program starts again.

Note: Before a program is downloaded, it must be edited to include the activation commands PROG:LEARN ACTIVE, PROG:LEARNLINE, and PROG:LEARN INACTIVE, which are described on page 1-3.

Program example

```
LABEL menu
    CLS
    CLRSTORE
    LPRINT "Select a program number using data keys"
    LPRINT "1. Program 1"
    LPRINT "2. Program 2"
    LPRINT "3. Program 3"
    LPRINT "4. Program 4"

LABEL start
    GETKEY char1
    IF char1 = 0 THEN
      GOTO stop
    ENDIF
    IF char1 < 81 | char1 > 84 THEN
      CLS
      CLRSTORE
      LPRINT "Invalid program number..."
      LPRINT "Please try again..."
      WAIT 1500
      GOTO menu
    ENDIF

    prog_num = char1 - 80

    CLRSTORE
    IF prog_num = 1 THEN
      GOSUB prog1
    ELSEIF prog_num = 2 THEN
      GOSUB prog2
    ELSEIF prog_num = 3 THEN
      GOSUB prog3
    ELSEIF prog_num = 4 THEN
      GOSUB prog4
    ENDIF
    WAIT 3000
    GOTO menu
LABEL stop
    CLRSTORE
    END

REM Subroutines go after the END statement
LABEL prog1
    LPRINT "This is program 1"
    RETURN

LABEL prog2
    LPRINT "This is program 2"
    RETURN

LABEL prog3
    LPRINT "This is program 3"
    RETURN

LABEL prog4
    LPRINT "This is program 4"
    RETURN
```

Downloading programs; PC to test set

To download a program from a PC to the test set, the test set and PC must be connected via an RS232 link or a GPIB link. The program is then downloaded using remote commands.

Note: The test set must have the appropriate system selected before a program is downloaded.

Two types of program can be executed on the test set; the built-in test routines or a user-defined program. After a user-defined program has been downloaded, the user must ensure that the USER DEFINED TEST program has been selected.

The test set only stores one MI-BASIC program at a time.

Three remote commands are needed to download a program. These are:-

```
PROG:LEARN ACTIVE
PROG:LEARNLINE
PROG:LEARN INACTIVE
```

A brief description of these commands follows.

PROG:LEARN ACTIVE

This command is sent first, as it clears the memory area used to store the user program. It also clears any relevant flags regardless of the state of the current system and indicates to the test set that a user program is about to be downloaded. If a `PROG:LEARNLINE` command is sent when `LEARN` is `INACTIVE` then an error is returned. On receipt of this command, the AUTORUN screen is displayed and the message **** LEARN MODE **** appears on the screen.

The status of `PROG:LEARN` can be found by using the `PROG:LEARN?` query command. Either `ACTIVE` or `INACTIVE` will be returned to the remote device.

PROG:LEARNLINE

This command downloads the program line by line. Each line of the program is enclosed within single quotes, and prefixed by `PROG:LEARNLINE`. Using the program example on the previous page, the first few lines of the program are sent as:-

```
PROG:LEARNLINE 'LABEL menu'
PROG:LEARNLINE '        CLS'
PROG:LEARNLINE '        CLRSTORE'
```

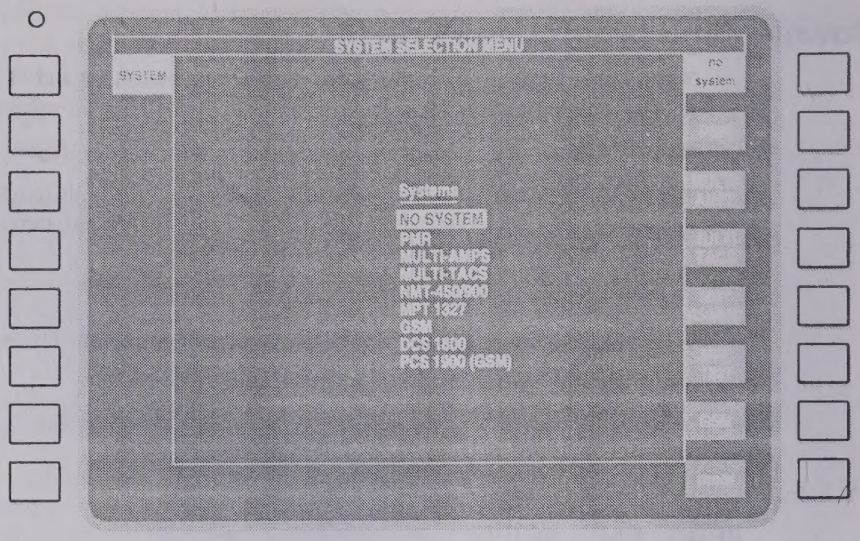
This is repeated for every line of the program.

PROG:LEARN INACTIVE

When all of the program has been downloaded, this command is sent to terminate the download process.

Step-by-step procedure for downloading a program

1. Write the program using any text editor in accordance with the statements and relevant syntax as described in Chapter 2. Edit the program to include the activation commands as described above. For the program example shown on page 1-2, the result would be as shown on page 1-5.
2. Link the PC and test set. Connect the PC to the test set via a GPIB link or an RS232 link.
3. Set up the PC and test set to receive and transmit using the correct remote controller. Press [HELP SET-UP], *[SET-UP], [INPUT OUTPUT], [remote options], [remote input]* to select either RS232 or GPIB. Then press [HELP SET-UP] followed by *[return]* to return to the previously-selected display.
4. Select the relevant system on the test set:-
   Press the [SYSTEMS] key; the display shown in Fig. 1-1 on page 1-4 appears. Select the relevant system, for example, GSM.



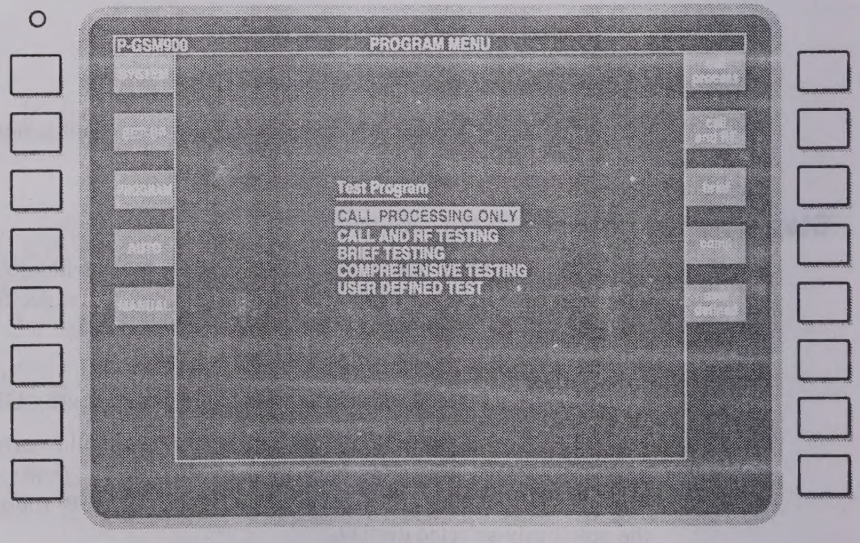
C3507

Fig. 1-1 System selection menu

5. Download the program.
   Transfer the program file to the test set (e.g. using Send Text File in a Windows Terminal, with Strip LF selected). The test set will automatically go into Auto mode, remote operation. During the download, the message **** LEARN MODE **** flashes on the screen

6. When the program has been downloaded, press the [HELP SET-UP] hard key to return to local operation.

7. Press the *[PROGRAM]* soft key; the display will be similar to that shown in Fig. 1-2. Then press the *[user defined]* soft key to select the downloaded program.
   Note: Pressing *[user defined]* has no effect unless a user-defined program is in the test set.



C2424

Fig. 1-2 Program menu

6. Run the program. The program can be run from the test set or by remote control.

   To run from the test set:-

   Press the *[start]* soft key.

   To run from remote, send the command:-

   ```
   PROG:RUNSTATE RUN
   ```

   Any errors with the syntax show only when the program is run for the first time. Syntax errors are shown in the results store and the *[start]* soft key (top right) is highlighted and labelled error. If there are any errors, edit the program and download it again.

   The run status of the test set can be checked from the remote device by using the `PROG:RUNSTATE?` query command. This will return either `STOP`, `RUN`, or `PAUSE`.

The program example shown on page 1-2 with activation commands included is shown below.

```
PROG:LEARN ACTIVE
PROG:LEARNLINE 'LABEL menu'
PROG:LEARNLINE '        CLS'
PROG:LEARNLINE '        CLRSTORE'
PROG:LEARNLINE '        LPRINT "Select a program number using data
keys"'
PROG:LEARNLINE '        LPRINT "1. Program 1"'
PROG:LEARNLINE '        LPRINT "2. Program 2"'
PROG:LEARNLINE '        LPRINT "3. Program 3"'
PROG:LEARNLINE '        LPRINT "4. Program 4"'
PROG:LEARNLINE ''
PROG:LEARNLINE 'LABEL start'
PROG:LEARNLINE '        GETKEY char1'
PROG:LEARNLINE '        IF char1 = 0 THEN'
PROG:LEARNLINE '              GOTO stop'
PROG:LEARNLINE '        ENDIF'
PROG:LEARNLINE '        IF char1 < 81 | char1 > 84 THEN'
PROG:LEARNLINE '              CLS'
PROG:LEARNLINE '              CLRSTORE'
PROG:LEARNLINE '              LPRINT "Invalid program number..."'
PROG:LEARNLINE '              LPRINT "Please try again..."'
PROG:LEARNLINE '              WAIT 1500'
PROG:LEARNLINE '              GOTO menu'
PROG:LEARNLINE '        ENDIF'
PROG:LEARNLINE ''
PROG:LEARNLINE '        prog_num = char1 - 80'
PROG:LEARNLINE ''
PROG:LEARNLINE '        CLRSTORE'
PROG:LEARNLINE '        IF prog_num = 1 THEN'
PROG:LEARNLINE '              GOSUB prog1'
PROG:LEARNLINE '        ELSEIF prog_num = 2 THEN'
PROG:LEARNLINE '              GOSUB prog2'
PROG:LEARNLINE '        ELSEIF prog_num = 3 THEN'
PROG:LEARNLINE '              GOSUB prog3'
PROG:LEARNLINE '        ELSEIF prog_num = 4 THEN'
PROG:LEARNLINE '              GOSUB prog4'
PROG:LEARNLINE '        ENDIF'
PROG:LEARNLINE '        WAIT 3000'
PROG:LEARNLINE '        GOTO menu'
PROG:LEARNLINE ''
PROG:LEARNLINE 'LABEL stop'
PROG:LEARNLINE '        CLRSTORE'
PROG:LEARNLINE '        END'
PROG:LEARNLINE ''
PROG:LEARNLINE 'REM Subroutines go after the END statement'
PROG:LEARNLINE 'LABEL prog1'
PROG:LEARNLINE '        LPRINT "This is program 1"'
PROG:LEARNLINE '        RETURN'
```

```
PROG:LEARNLINE ''
PROG:LEARNLINE 'LABEL prog2'
PROG:LEARNLINE '        LPRINT "This is program 2"'
PROG:LEARNLINE '        RETURN'
PROG:LEARNLINE ''
PROG:LEARNLINE 'LABEL prog3'
PROG:LEARNLINE '        LPRINT "This is program 3"'
PROG:LEARNLINE '        RETURN'
PROG:LEARNLINE ''
PROG:LEARNLINE 'LABEL prog4'
PROG:LEARNLINE '        LPRINT "This is program 4"'
PROG:LEARNLINE '        RETURN'
PROG:LEARN INACTIVE
```

Storing and recalling programs on memory cards

MI-BASIC programs can be stored externally on a memory card inserted in the memory card slot. See any of the System Supplements for details of how to store and recall programs on a memory card.

Chapter 2
MI-BASIC REFERENCE GUIDE

Contents

Syntax information 2-3
Operators 2-7
Numeric operators 2-8
String operators 2-11
Commands 2-12
NUMRESULTS error codes 2-36
Reading special test parameters 2-42
Setting special test parameters 2-57
Instrument key decimal values 2-58

List of commands

ACCESSORY 2-12
AFGEN*n* FREQ 2-13
AFGEN*n* LEVEL 2-14
AFGEN*n* STATUS 2-14
ATOF 2-15
CLEAR 2-15
CLRSTORE 2-16
CLS 2-16
DATAFCC 2-17
DEFAULT 2-17
DEMODTYPE 2-17
DISPLAY CLEAR 2-18
DISPLAY PRINTFR 2-18
DISPLAY PRINTOHD 2-19
END 2-19
GETCOMPANY 2-20
GETDATE 2-20
GETINSTID 2-20
GETKEY 2-21
GETPAUSE 2-22
GETRXFREQ 2-23
GETSYS 2-23
GETTIME 2-23
GETTXFREQ 2-23
GETUSER 2-24
GOSUB 2-24
GOTO 2-25
IF, THEN, ELSEIF, ELSE, ENDIF 2-26
LABEL 2-27
LASTKEY 2-27
LET 2-28
LPRINT 2-29
MEASURE 2-31
MODGEN*n* FREQ 2-32
MODGEN*n* LEVEL 2-33
MODGEN*n* STATUS 2-34
MODTYPE 2-34
NUMRESULTS 2-35

PRINT AT 2-37
PRINTF AT 2-38
PRINTOFF 2-39
PRINTON 2-40
READPARAM 2-41
RECEIVER AUTOTUNE 2-43
RECEIVER CHAN 2-43
RECEIVER FREQ 2-43
REM 2-44
RETURN 2-44
RFGEN CHAN 2-45
RFGEN FREQ 2-45
RFGEN LEVEL 2-46
RFGEN STATUS 2-46
RFPORT 2-47
RS232_IN 2-48
RS232_OUT 2-49
RXFILTER 2-49
SOFTKEY 2-50
SPRINT 2-51
STRRESULTS 2-52
TEST 2-53
TXFILTER 2-54
WAIT 2-55
WRITEPARAM 2-56

List of figures

Fig. 2-1 Accessory port plug contacts (as seen facing panel) 2-12
Fig. 2-2 Co-ordinate reference points on the display 2-15
Fig. 2-3 Front panel layout 2-61
Fig. 2-4 Key code number definition 2-61

List of tables

Table 2-1 Key descriptions versus decimal values 2-58
Table 2-2 Decimal values versus key descriptions 2-59

<num>

Description: A number. This can be in the range $1.797 \times 10^{+308}$ to 2.225×10^{-308}. Only decimal notation is supported.
See also *Numeric operators* (page 2-8).

Examples: Valid numeric entries are:-

```
6, 12, 0.34, .56, 123.45e-12, 1.0e003
```

<num var>

Description: A numeric variable. This is a name for a number that varies. Maximum variable name length is 24 characters. See also *Numeric operators* (page 2-8).

Examples: In these examples, `fred` and `tom` are <num var>s:-

```
fred = 12
tom = 1.07e2
```

<num expr>

Description: A numeric expression. This is a mathematical expression that has <num> and / or <num var> for its terms, which are related by numerical operators. In its simplest form, a numeric expression may be a number or a numeric variable.
See also *Numeric operators* (page 2-8).
In this example, `harry`, `fred` and `tom` are <num var>s, `3` is a <num>, and `+` and `*` are numerical operators. The whole line constitutes a <num expr>.

Example:

```
harry = (fred + 3) * tom
```

<str>

Description: A string. This is a line of ASCII characters and is denoted by being enclosed within quotes - `" "`.
See also *String operators* (page 2-11).

Examples:

```
"Hello world"
"It's raining again!"
```

<str var>

Description: A string variable. This is a name for a string that varies. Maximum variable name length is 24 characters.
A string variable is distinct from a numeric variable in that it always ends in a `$` sign.
See also *String operators* (page 2-11).

Examples:

```
hello$ = "Hello world"
again$ = "It's raining again!"
```

<str expr>

Description: A string expression. This is an expression that has <str> and / or <str var> for its terms, which are related by string operators.
See also *String operators* (page 2-11).
In the examples of string expressions below, `hello$`, `again$`, `first$`, `world$`, and `raining$` are <str var>s, `"again"` and `"world"` are <str>s, `+` and `-` are string operators.

Example:
```
Let hello$ = "Hello world"
Let again$ = "It's raining again!"
If first$ = again$ - " again" then
    first$  = "It's raining!"          (extraction)
If world$ = hello$ - "world" then
    world$ = "Hello "                  (extraction)
If raining$ = world$ + first$ then
    raining$ = "Hello It's raining!" (concatenation)
```

CAPITAL LETTERS

Description: A command keyword. **BOLD CAPITAL LETTERS** denote a command keyword in this chapter.

Example: **DEFAULT**

<num *n*>
<num var *n*>
<str *n*>
<str var *n*>

Description: These names are used in command statements requiring more than one item to indicate the type and number of possible values.
If *n* is missing, it implies one item.
In the example below, the keyword **NUMRESULTS** must be followed by a <num var>. A further seven <num var>s are optional.

Syntax: **NUMRESULTS** <num var1> [,...,<num var8>]

Example:
```
NUMRESULTS result, act, tgt
```

<name>

Description: Identifier for a keyword, it is used in program control. Maximum length is 24 characters. Invalid <name>s are keywords, numbers, string variables, strings.

Syntax: **LABEL** <name>

Example:
```
LABEL FRED
```

<params>

Description: Test parameters defined in Chapter 3 and cross referenced to relevant manuals.

[ ]

Description: An option. Anything enclosed within square brackets is an optional part of the command statement.

This example shows that the keyword **GETKEY** can be used by itself or it can have a <num var> after it.

Syntax: **GETKEY** [<num var>]

Examples:
```
GETKEY
GETKEY escape
```

| |

Description: A choice. Parallel lines represent a choice; one of the terms within the parallel lines must be chosen, unless the parallel lines are enclosed within square brackets.

In this example, after the keyword **SOFTKEY**, you must choose either a <num> or a <num var>; after that you must also choose from **NORMAL**, **BRIGHT**, **DELETE** or **CLEAR**. There is then a choice between using <str1> or <str1> <str2>, but this is an optional part of the command statement.

Syntax: **SOFTKEY** | <num> / <num var> | , | NORMAL / BRIGHT / DELETE / CLEAR | [| <str1> / <str1><str2> |]

Examples:
```
SOFTKEY 2, NORMAL "FRED"
SOFTKEY 7, BRIGHT "test" "start"
SOFTKEY 33, DELETE
fred = 34
SOFTKEY fred, CLEAR
```

...

Description: Repeat. Previous item(s) may be repeated as necessary.

In this example, the keyword **STRRESULTS** must be followed by a <str var>. A further seven <str var>s are optional.

Syntax: **STRRESULTS** <str var1> [, ... ,<str var8>]

Examples:
```
STRRESULTS res1$
STRRESULTS title$, stat$, comment1$
```

<statement>

Description: A command statement. An MI-BASIC expression which includes a command keyword and relevant arguments.

In this example, the keyword **GETKEY** is used with a <num var> called `choice`. When a key is pressed, its decimal value is put into the <num var> `choice`.

Example:
```
GETKEY choice
```

<program segment>

Description: A program segment. Two or more <statement>s.
In the example, **GETKEY** stops the program until a key is pressed; the decimal value of the key is returned into `choice`. If the value of the key pressed is 2, then the program will print choice = 2 on the screen, otherwise the program will continue.

Example:
```
GETKEY choice
IF choice = 2 THEN
    LPRINT "choice = 2"
ENDIF
```

<program>

Description: A program. A program is one or more <statements> that are executed in sequence.
This example program fills the screen with the * character. The screen is filled one character at a time, column by column.

Example:
```
    start_column = 0
    start_row = 0
    column = start_column
    row = start_row
LABEL start
    PRINT AT column,row; "*"
    row = row +1

    IF row > 26 THEN
        column = column + 1
        row = start_row
    ENDIF

    IF column > 45 THEN
        GOTO stop
    ENDIF

GOTO start

LABEL stop
END
```

Operators

Parenthesis

Expressions can contain a high degree of complexity. Parentheses are allowed to any level and should be used whenever there is a possibility of ambiguity.

Precedence

The table below lists the operators in groups of equal precedence, the highest priority group first.

Symbol	*Description*
!	NOT
( )	Brackets
+	Plus, [Unary]
–	Minus, [Unary]
LOG	Natural logarithm
LOG10	Base 10 logarithm
EXP	Natural exponent
EXP10	Base 10 exponent
ABS	Absolute value
/	Divide
*	Multiply
MOD	Modulus or remainder
+	Plus, [Binary]
-	Minus, [Binary]
<<	Left shift
>>	Right shift
>=	Greater than or equal to
>	Greater than
<=	Less than or equal to
<	Less than
=	Equal to
<>	Not equal to
&	Bitwise AND
^	Bitwise XOR
|	Bitwise OR
AND	AND
XOR	XOR
OR	OR

Numeric operators

Arithmetic operators (binary)

Operator	*Description*
+	Plus
–	Minus
/	Divide
*	Multiply
MOD	Modulus or remainder

Example The function of MOD is to give the remainder of the first value divided by the second value:-

```
If   result1 = 25 MOD 10 then
     result1 = 5

If   result2 = 25 MOD 5 then
     result2 = 0

If   result3 = 351 MOD 251 then
     result3 = 100
```

Arithmetic operators (unary)

Operator	*Description*
+	Plus.
–	Minus.
LOG	Natural logarithm.
LOG10	Base 10 logarithm.
EXP	Natural exponent
EXP10	Base 10 exponent
ABS	Absolute value

Conditional operators

Operator	*Description*
=	Equal to
< >	Not equal to
> =	Greater than or equal to
>	Greater than
< =	Less than or equal to
<	Less than

Bitwise operators (binary)

~

Description: Negate (Bitwise NOT, unary)

Example:
```
If   result = ~(-101) then
     result = 100
```

Explanation:

$$-101_{10} = 1\ 1001\ 1011_{2}$$
$$\text{invert} = 0\ 0110\ 0100_{2}$$
$$0\ 0110\ 0100_{2} = 64_{16}$$
$$64_{16} = 100_{10}$$
$$\therefore \text{result} = 100$$

Bitwise operators (unary)

Operator

<<

Description: Left shift

Example:
```
If   result = 25 << 2 then
     result = 100
```

Explanation:

$$25_{10} = 11001_2$$
$$11001_2 << 2 = 1100100_2$$
$$1100100_2 = 100_{10}$$
$$\therefore \text{result} = 100 \text{ (i.e. } 25 \times 4)$$

>>

Description: Right shift

Example:
```
If   result = 400 >> 2 then
     result = 100
```

Explanation:

$$400_{10} = 110010000_2$$
$$110010000_2 >> 2 = 1100100_2$$
$$1100100_2 = 100_{10}$$
$$\therefore \text{result} = 100 \text{ (i.e. } 400 \div 4)$$

&

Description: Bitwise AND

Example:
```
If   result = 255 & 100 then
     result = 100
```

Explanation:

$$255_{10} = 1111\ 1111_2$$
$$100_{10} = 0110\ 0100_2$$
$$1111\ 1111_2 \,\&\, 0110\ 0100_2 = 0110\ 0100_2$$
$$0110\ 0100_2 = 100_{10}$$
$$\therefore \text{result} = 100$$

|

Description: Bitwise OR

Example:
```
If   result = 96 | 4 then
     result = 100
```

Explanation:

$$96_{10} = 110\ 0000_2$$
$$4_{10} = 000\ 0100_2$$
$$110\ 0000 \mid 000\ 0100_2 = 110\ 0100_2$$
$$110\ 0100_2 = 100_{10}$$
$$\therefore \text{result} = 100$$

^

Description: Bitwise XOR

Example:
```
If   result = 255 ^ 155 then
     result = 100
```

Explanation:

$$255_{10} = 1111\ 1111_2$$
$$155_{10} = 1001\ 1011_2$$
$$1111\ 1111_2 \,\hat{}\, 1001\ 1011_2 = 0110\ 0100_2$$
$$0110\ 0100_2 = 100_{10}$$
$$\therefore \text{result} = 100$$

Logical operators (binary)

Operator

AND

Description: AND

Example: If `result = 255 AND 100 then`
`result = 1`

Explanation:

255 = non-zero
non-zero = true
true = 1

100 = non-zero
non-zero = true
true = 1

1 AND 1 = 1
∴ result = 1

OR

Description: OR

Example: If `result = 96 OR 4 then`
`result = 1`

Explanation:

96 = non-zero
non-zero = true
true = 1

4 = non-zero
non-zero = true
true = 1

1 OR 1 = 1
∴ result = 1

XOR

Description: XOR

Example: If `result = 96 XOR 4 then`
`result = 0`

Explanation:

96 = non-zero
non-zero = true
true = 1

4 = non-zero
non-zero = true
true = 1

1 XOR 1 = 0
∴ result = 0

Logical operators (unary)

Operator

!

Description: NOT

Example:
```
If   result = !(0) * 100 then
     result = 100
```

Explanation:

$$!(0) = 1$$
$$1 * 100 = 100$$
$$\therefore \text{result} = 100$$

String operators

Operator

+

Description: Plus (concatenate)

Example:
```
Let   hello$ = "Hello"
and   world$ = " world"

If    result$ = hello$ + world$
then  Lprint result$
```
will print the string `Hello world`

–

Description: Minus (search and remove)

Example:
```
Let   helloworld$ = "Hello world"
and   world$ = " world"

If     result$ = helloworld$ - world$
then   Lprint result$
```
will print the string `Hello`

ACCESSORY

Function: To set the accessory port's logic levels high or low.

Syntax: **ACCESSORY** | LOGIC0 | LOGIC1 | LOGIC2 | LOGIC3 | | HIGH | LOW |

Remarks: LOGIC0, LOGIC1, LOGIC2 and LOGIC3 are contacts on the ACCESSORY socket on the rear panel of the instrument.

Example: This command statement sets the accessory port's LOGIC1 level high:-

```
ACCESSORY LOGIC1 HIGH
```

Fig. 2-1 Accessory port plug contacts (as seen facing panel)

The table below is included for information only. No inference is made as to the uses to which this connector can be used for connecting any equipment other than approved devices manufactured by Marconi Instruments. The functions of the plug contacts are as shown below.

Pin No.	Function	Remarks
1	No connection	For future expansion. Do not use.
2	No connection	For future expansion. Do not use.
3	+12 V	100 mA max.
4	Ground (Logic)	Use with power accessory lines.
5	Acc-SDA	I^2C data.
6	Status 2	Used by 2955 accessories.
7	Logic 3	See Note 1 below.
8	Logic 1	See Note 1 below.
9	Speaker	Maximum load 8 Ω.
10	No connection	
11	Ground (REV PWR)	Used by directional power head.
12	Ground (FWD PWR)	Used by directional power head.
13	No connection	For future expansion. Do not use.
14	No connection	For future expansion. Do not use.
15	No connection	For future expansion. Do not use.
16	–12 V	100 mA max.
17	Acc SCL	I^2C clock.
18	Status 3	Used by 2955 accessories.
19	Status 1	Used by 2955 accessories.
20	Logic 2	See Note 1 below.
21	Logic 0	See Note 1 below.
22	Ground (SPKR)	Speaker ground.
23	REV PWR	Used by directional power head.
24	FWD PWR	Used by directional power head.
25	No connection	For future expansion. Do not use.

Note 1

Programmable in SYSTEMS MODE.

Open drain drive pulled up to +5 V with 4.7 kΩ. Max sink current 10 mA for V_{OL} = 1 V.

AFGEN*n* FREQ

Function: To set the designated AF generator's frequency.

Syntax: **AFGEN*n* FREQ** <num> | <num var> | <num expr> [Hz | kHz]

If the unit is omitted, the default unit is Hz.

Remarks: The *n* represents which generator is to be set, where *n* = 1, 2 or 3. The frequency is set using either a number, a numeric variable or a numeric expression. The unit is optional and is set using Hz or kHz. The frequency range for each generator is 10 Hz to 20 kHz.

This command is equivalent to using the AF GEN, FREQ and DATA entry keys.

Example: All of these examples assume that the AF generators are ON. This can be done from MI-BASIC using the following command statement, where *n* = 1, 2 or 3:-

```
AFGENn STATUS ON
```

Also, the level of the designated generator must be non-zero and the sum of all three generator levels must be less than 5 V.

1. This command statement uses a number for the frequency without specifying the optional units and sets AF generator 1 frequency to 100 Hz.

   ```
   AFGEN1 FREQ 100
   ```

2. This program segment sets AF generator 2 frequency to 5 kHz by using a numeric variable for the frequency and specifying the unit to be kHz.

   ```
   value = 5
   AFGEN2 FREQ value kHz
   ```

3. This program segment sets AF generator 3 frequency to 6 kHz by using a numeric expression for the frequency and specifying the unit to be kHz. This command may be part of a loop where the frequency is incremented by 1 kHz each time the loop is passed through.

   ```
   value = 5
   AFGEN3 FREQ value + 1 kHz
   ```

AFGEN*n* LEVEL

Function: To set the designated AF generator's level.

Syntax: **AFGEN*n* LEVEL** <num> | <num var> | <num expr> [UVOLT | MVOLT | VOLT]

The default unit is V.

Remarks: The *n* represents which generator is to be set, where *n* = 1, 2 or 3. The level is set using either a number, a numeric variable or a numeric expression. The level must be non-zero and the sum of all the AF generator levels must be less than 5 V.

This command is equivalent to using the AF GEN, LEVEL and DATA entry keys.

Example: All of these examples assume that the AF generators are ON. This can be done from MI-BASIC using the following command statement, where *n* = 1, 2 or 3:-

```
AFGENn STATUS ON
```

1. This command statement uses a number for the level and sets the AF generator 1 level to 1 V.

```
AFGEN1 LEVEL 1
```

2. This program segment uses a numeric variable for the level and sets the AF generator 2 level to 0.5 V.

```
value = 0.5
AFGEN2 LEVEL value
```

3. This program segment uses a numeric expression for the level and sets the AF generator 3 level to 3.5 V. The command statement that contains the AFGEN command may be part of a loop where the level is incremented by 0.5 V each time the loop is passed through.

```
value = 3
AFGEN3 LEVEL value + 0.5
```

AFGEN*n* STATUS

Function: To switch the designated AF generator ON or OFF.

Syntax: **AFGEN*n* STATUS** ON | OFF

Remarks: The *n* represents which generator is to be set, where *n* = 1, 2 or 3. This command is equivalent to using the *[gen n on OFF]* soft keys.

Example: 1. This command statement switches AF generator 1 ON:-

```
AFGEN1 STATUS ON
```

2. This command statement switches AF generator 2 OFF:-

```
AFGEN2 STATUS OFF
```

ATOF

Function: To convert a string variable into a numeric variable.

Syntax: **ATOF** (<num var>,<str var>)

Remarks: The **SPRINT** command does the opposite, it converts a numeric variable into a string variable.

Example: This program segment converts a string variable `value$` into a numeric variable `fred` and then prints the result to the screen.

The characters 1234.000000 are printed to the screen.

```
value$ = "1234"
ATOF(fred,value$)
LPRINT fred
```

CLEAR

Function: To clear an area of the screen.

Syntax: **CLEAR(** <num1> | <num var1> | <num expr1> **,** <num 2> | <num var2> | <num expr2> **,** <num 3> | <num var3> | <num expr3> **,** <num 4> | <num var4> | <num expr4> **)**

Remarks: The test set screen size is 640 × 480 pixels, with the origin at the top left corner. The **CLEAR** command clears a rectangular area of the screen. The area to be cleared is defined by coordinates for the top left corner (x1,y1) and for the bottom right corner (x2,y2). Putting the coordinates in the CLEAR command statement gives:-

```
CLEAR(x1,y1,x2,y2)
```

Example: The following example clears an area of the screen that has corner coordinates as follows:-

Top left corner (160,120)
Top right corner (480,120)
Bottom left corner (160,360)
Bottom right corner (480,360)

This command statement uses a combination of numbers, a numeric variable and a numeric expression for the coordinates.

```
coord2 = 120
CLEAR(160,coord2, 4*coord2,360)
```

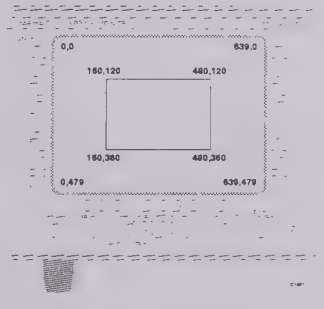


Fig. 2-2 Co-ordinate reference points on the display

CLRSTORE

Function: To clear the currently displayed results store.

Syntax: **CLRSTORE**

Remarks: This command is equivalent to the *[clear store]* soft key.

Example: This program segment prints `Hello World!!` into the results store, waits for 3 seconds and then clears the store:-

```
LPRINT "Hello World!!"
WAIT 3000
CLRSTORE
```

CLS

Function: To clear the display except for the area allocated to the soft keys.

Syntax: **CLS**

Example: This program segment fills the screen with the * character, waits for 5 seconds and then uses the **CLS** command to clear the middle of the screen:-

```
      start_column = 0
      start_row = 0
      column = start_column
      row = start_row
LABEL start
      PRINT AT column,row; "*"
      row = row +1

      IF row > 26 THEN
            column = column + 1
            row = start_row
      ENDIF

      IF column > 45 THEN
            GOTO stop
      ENDIF

      GOTO start

LABEL stop
      WAIT 5000
      CLS
      END
```

DATAFCC

Function: To set-up the test set to initialise and start Forward Control Channel (FCC) signalling.

Syntax: **DATAFCC**

Remarks: This command may only be required with signalling systems that use a forward control channel, when it should be executed at the beginning of a test program.

Example: This program segment is at the beginning of a test program. The **DATAFCC** command clears the pass / fail count flags, then initialises and starts forward control channel signalling. **CLRSTORE** clears the results store.

```
REM start of a test program
DATAFCC
CLRSTORE
```

DEFAULT

Function: To reset the test set's parameters for the current system variant.

Syntax: **DEFAULT**

Remarks: If a signalling system is in use, the **DEFAULT** command also re-initialises and restarts the signalling system and resets the pass / fail count flags.

Example: This program segment is at the beginning of a test program; it sets everything up prior to testing. The **DEFAULT** command clears the pass / fail count flags, resets the test set to its default parameters, and then initialises and starts forward overhead signalling. **CLRSTORE** clears the results store.

```
REM start of a test program
DEFAULT
CLRSTORE
```

DEMODTYPE

Function: Sets the receiver to expect a particular type of modulation.

Syntax: **DEMODTYPE** | AM | FM | PM |

Remarks:

Example: This command statement sets the receiver to demodulate AM signals:-

```
DEMODTYPE AM
```

DISPLAY CLEAR

Function: To clear the data display.

Syntax: **DISPLAY CLEAR**

Remarks: This command is not applicable in GSM, DCS, PCS or PMR; it is used to clear the forward / reverse data from the screen (screen titled FORWARD / REVERSE DATA). It is equivalent to pressing the *[clear]* soft key while in that screen.

Example: This program segment is at the beginning of a program. The **DEFAULT** command resets the test set and clears the pass / fail count flags; **DISPLAY CLEAR** clears the data display.

```
REM start of a program
DEFAULT
DISPLAY CLEAR
```

DISPLAY PRINTFR

Function: To transfer the expanded forward / reverse data display's data into the results store.

Syntax: **DISPLAY PRINTFR**

Remarks: This command is used during a signalling test such as AMPS or TACS (not GSM, DCS, PCS or PMR) to put the data into the results store.

Example: This program segment executes a **TEST** command; if the test fails, indicated by the fail flag being non zero, the forward / reverse data display's data is transferred into the results store.

Once the display data is in the results store, it can be printed out.

```
TEST PLACECALL
NUMRESULTS error_flag
IF error_flag <> 0 THEN
      DISPLAY PRINTFR
ENDIF
```

DISPLAY PRINTOHD

Function: To transfer the expanded overhead data display's data into the results store.

Syntax: **DISPLAY PRINTOHD**

Remarks: This command is used during a signalling test such as AMPS or TACS (not GSM, DCS, PCS or PMR) to access the data.

Example: This program segment executes a **TEST** command; if the test fails, indicated by the fail flag being non-zero, the overhead data display's data is transferred into the results store.

Once the display's data is in the results store, it can be printed out.

```
TEST PLACECALL
NUMRESULTS error_flag
IF error_flag <> 0 THEN
      DISPLAY PRINTOHD
ENDIF
```

END

Function: To terminate the program's execution.

Syntax: **[END]**

Remarks: The **END** command is optional and can be used at any point in a program, it does not necessarily have to be at the end of an input file.

The **END** command can only be used once in a program.

To improve program readability, it is recommended that all subroutines are positioned after the **END** command.

Example: This program has a subroutine to print a message to the screen. The program goes around the loop 10 times; each time it clears the results store and then prints the message in the results store. After 10 loops, the program terminates.

The subroutine is positioned after the **END** command.

```
      count = 0
LABEL start
      count = count + 1
      GOSUB message

      IF count < 10 THEN
            GOTO start
      ENDIF

      END

LABEL message
      CLRSTORE
      LPRINT "Hello"
      LPRINT "Count = ",USING "2.0" count
      RETURN
```

GETCOMPANY

Function: To get the name of the Company.

Syntax **GETCOMPANY** <str var>

Remarks: The Company name can be entered via the HELP AND SET-UP menu, as explained in the Operating Manual. If a name has not been entered, a string of full stops is returned.

Example:

```
GETCOMPANY company$
LPRINT "COMPANY:   ";company$
```

GETDATE

Function: To get the current date from the real-time clock.

Syntax: **GETDATE** <str var>

Remarks: The date format is *dd/mm/yy*.

Example:

```
GETDATE date$
LPRINT "DATE:   ";date$
```

GETINSTID

Function: To get the instrument's serial number.

Syntax: **GETINSTID** <str var>

Example:

```
GETINSTID instid$
LPRINT "INSTRUMENT:   ";instid$
```

GETKEY

Function: To pause program execution until a key is pressed and optionally return the decimal value of the key pressed.

Syntax: **GETKEY** [<num var>]

Remarks: The program execution is halted until any key is pressed. If <num var> is specified, the decimal value of the pressed key is returned into the <num var>. Tables 2-1 and 2-2, at the end of this chapter, cross-refer the keys to their decimal values. Figures 2-3 and 2-4, also at the end of this chapter, show the key locations with decimal values overwritten.

Example: This program segment draws up 3 soft keys labelled 1, 2 and 3, and prints a request for one of them to be selected. The **GETKEY** command is used with a <num var> called `choice`. The decimal value of the soft key pressed is returned into `choice`, and the value of `choice` determines which message is printed.

If a soft key other than 36, 37 or 38 is pressed, the loop is repeated, and the user is requested by a message printed to the screen to select soft key 1, 2 or 3 again. Once 1, 2 or 3 has been selected, and the relevant message printed to the screen, the program continues.

```
LABEL choose
      SOFTKEY 36,NORMAL "1"
      SOFTKEY 37,NORMAL "2"
      SOFTKEY 38,NORMAL "3"
      LPRINT "Please select 1,2 or 3"
      GETKEY choice

      IF choice = 36 THEN
            LPRINT "Key pressed was soft key 1"
      ELSEIF choice = 37 THEN
            LPRINT "Key pressed was soft key 2"
      ELSEIF choice = 38 THEN
            LPRINT "Key pressed was soft key 3"
      ELSE
            GOTO choose
      ENDIF
```

Function: To pause program execution.

Syntax: **GETPAUSE** [<num> | <num var> | <num expr>]

Remarks: There are different pause modes that can be set from either the AUTORUN CONTROL menu or from remote control. They are:

ALWAYS	The program execution is paused every time the **GETPAUSE** command occurs.
MANUAL ONLY	The program execution is only paused if the pause key has been pressed, otherwise the **GETPAUSE** command is ignored.
ON FAILURE	When the **GETPAUSE** command follows a test and the test fails, the program is paused.

If the optional <num> / <num var> / <num expr> is used, it must be set to 1 or 0. If it is set to 1, the program pauses at the **GETPAUSE** command regardless of which pause mode is set.
If the <num> / <num var> / <num expr> = 0, the **GETPAUSE** command is ignored regardless of which pause mode is set.

Example: In the following example, **GETPAUSE** is used by itself and with an optional numeric variable.

Pause mode MANUAL ONLY (OFF). When this pause mode is set, if the pause key is pressed at the beginning of this program segment, the program pauses at the first **GETPAUSE** (after printing `hello...`). If the pause key is not pressed, the program ignores the first **GETPAUSE**. The program stops at the second **GETPAUSE** (after printing `1...`), as the numeric variable is set to 1. The third **GETPAUSE** is ignored, as the numeric variable is set to 0.

Pause mode ALWAYS. When this pause mode is set, the program segment pauses at the first **GETPAUSE** and the second **GETPAUSE**. It ignores the third **GETPAUSE**, as the numeric variable is set to 0.

```
LPRINT "hello..."
GETPAUSE
LPRINT
LPRINT "1..."
fred = 1
GETPAUSE fred
LPRINT "2..."
fred = 0
GETPAUSE fred
LPRINT "3..."
```

GETRXFREQ

Function: To return the current frequency of the RF generator/radio's receiver.

Syntax: **GETRXFREQ** <num var>

Remarks: If the frequency of a particular channel needs to be read, the RF generator must be set to that channel first, using the **RFGEN CHAN** command (PMR only).

Example: The following program segment reads the frequency of channel 9. The RF generator frequency is set to that of channel 9 and then the frequency of the RF generator is read and printed to the results store.

```
channel = 9
RFGEN CHAN channel
GETRXREQ rx_freq
LPRINT rx_freq
```

GETSYS

Function: To get the current system name as a string variable.

Syntax: **GETSYS** <str var>

Example: This example gets the current system name and prints it to the results store.

```
GETSYS system$
LPRINT "SYSTEM: ";system$
```

GETTIME

Function: To get the time from the real-time clock.

Syntax: **GETTIME** <str var>

Remarks: The time format is *hh/mm/ss*.

Example:

```
GETTIME time$
LPRINT "TIME: ";time$
```

GETTXFREQ

Function: To return the current instrument receiver frequency/mobile transmitter frequency.

Syntax: **GETTXFREQ** <num var>

Remarks: If the frequency of a particular channel needs to be read, the receiver must be set to that channel first using the **RECEIVER CHAN** command (PMR only).

Example: The following program segment reads the frequency of channel 10. The channel is set to 10 and then the Tx frequency is read and printed to the results store.

```
channel = 10
RECEIVER CHAN channel
GETTXFREQ tx_freq
LPRINT tx_freq
```

GETUSER

Function: To get the name of the current user.

Syntax **GETUSER** <str var>

Remarks: The current user's name can be entered via the HELP AND SET-UP menu, as explained in the Operating Manual. If a name has not been entered, a string of full stops is returned.

Example:

```
GETUSER user$
LPRINT "TESTER:   ";user$
```

GOSUB

Function: To jump to a subroutine with the same identifier and return to the next line when execution of the subroutine has been completed.

Syntax: **GOSUB** <name>

Remarks: To improve program readability, it is recommended that the subroutines are placed after the **END** command. The maximum length for <name> is 24 characters. All subroutine <name>s should be unique; <name>s cannot be keywords, numbers, string variables or strings.

Example: This program has a subroutine to print a message on the screen. The program goes around the loop 10 times; each time it clears the results store and then prints the message in the results store. After 10 loops, the program terminates.

The subroutine is positioned after the **END** command.

```
      count = 0
LABEL start
      count = count + 1
      GOSUB message

      IF count < 10 THEN
            GOTO start
      ENDIF

      END

LABEL message
      CLRSTORE
      LPRINT "Hello"
      LPRINT "Count = ",USING "2.0" count
      RETURN
```

GOTO

Function: To jump to a **LABEL** with the same identifier.

Syntax: **GOTO** <name>

Remarks: Maximum length for <name> is 24 characters. All <names> should be unique; <name>s cannot be keywords, strings, string variables or numbers.

Example: This program has a subroutine to print a message on the screen. The program goes around the loop 10 times, using the **GOTO** start statement to go back to the beginning of the loop. Each time around the loop, it clears the results store and then prints the message in the results store. After 10 loops, the program terminates.

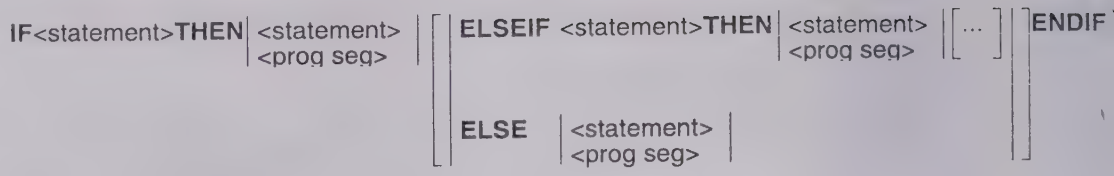
The subroutine is positioned after the **END** command.

```
      count = 0
LABEL start
      count = count + 1
      GOSUB message

      IF count < 10 THEN
            GOTO start
      ENDIF

      END

LABEL message
      CLRSTORE
      LPRINT "Hello"
      LPRINT "Count = ",USING "2.0" count
      RETURN
```

IF, THEN, ELSEIF, ELSE, ENDIF

Function: To provide control of program flow.

Syntax:



Remarks: Single-line **IF** statements must *not* contain **ENDIF**. An extended **IF** statement containing multiple lines must be terminated with an **ENDIF** (see example below).

Example: This example program asks the user to select a number from 1 to 3 using the data keys. It uses an **IF**, **THEN**, **ENDIF** command to check whether the number returned was valid; if it was, the program continues, if not, the program prints a message saying Number out of range and then returns to the start.

The decimal values of the numeric DATA keys are from 80 to 89, therefore an offset of 80 is used.

The program uses an **IF**, **THEN**, **ELSEIF**, **ELSE**, **ENDIF** command to determine which message to print on the screen.

```
        offset = 80
LABEL start
        CLRSTORE
        LPRINT "Please select a number from 1 to 3"
        LPRINT "using the data keys."
        GETKEY choice

        IF choice < 81 | choice > 83 THEN
              LPRINT "Number out of range"
              LPRINT "Please try again"
              WAIT 3000
              GOTO start
        ENDIF

        selected = choice - offset

        IF selected = 1 THEN
              GOSUB message1
        ELSEIF selected = 2 THEN
              GOSUB message2
        ELSE
              GOSUB message3
        ENDIF

        END

LABEL message1
        LPRINT "You have selected number 1"
        RETURN
LABEL message2
        LPRINT "You have selected number 2"
        RETURN
LABEL message3
        LPRINT "You have selected number 3"
        RETURN
```

LABEL

Function: To mark a position in a program to jump to from a **GOTO** or a **GOSUB**.

Syntax: **LABEL** <name>

Remarks: Maximum length for <name> is 24 characters; invalid <name>s are numbers, strings, string variables or keywords.

Example: In this program segment, the keyboard buffer is checked and equates a key press to the stored value. If no keys have been pressed, keypress = –1 and the program loops back to the **LABEL** start. If a key has been pressed, the program continues.

```
LABEL start
     LASTKEY keypress
     IF keypress = -1 THEN
      GOTO start
     ENDIF
```

LASTKEY

Function: To get the value of the last key pressed.

Syntax: **LASTKEY** <num var>

Remarks: This is a method of reading the key values without stopping the program. As a key is pressed, its value is stored in a numeric variable which is accessed by this keyword. This means that the value of the last key pressed can be read. Once this is called, the numeric variable is reset to –1 until the next key is pressed. Tables 2-1 and 2-2, at the end of this chapter, cross-refer the keys to their decimal values. Figures 2-3 and 2-4, also at the end of this chapter, show the key locations with decimal values overwritten.

Example: In this program segment, the keyboard buffer is checked and equates a key press to the value in the keyboard buffer. If no keys have been pressed, keypress = –1 and the program loops back to the start. If a key has been pressed, the decimal value of that key is printed to the screen and the program loops back to the start.

```
LABEL start
     CLRSTORE
     LPRINT "Please press a key..."
     WAIT 3000
     LASTKEY keypress
     value = keypress
     WAIT 5000
     IF keypress = -1 THEN
      GOTO start
     ELSE
      LPRINT "Key pressed was " value
     ENDIF
WAIT 3000
GOTO start
```

Function: To assign a numeric expression to a number or numeric variable or numeric expression or to equate a string variable to a string or string variable or string expression.

Syntax:

[LET] | <num var> = | <num> / <num var> / <num expr> |

| <str var> = | <str> / <str var> / <str expr> |

Remarks: **LET** is an optional keyword.

Example: The following pairs of code produce identical results:-

```
fred = 6
LET fred = 6

fred$ = "hello"
LET fred$ = "hello"
```

LPRINT

Function: To print to the current results store.

Syntax: **LPRINT** [<option> [,] <option> [,] ... [;]]

<option> = [**AT** | <num> / <num var> / <num exp> |] | <num> / <num var> / <num exp> / <str> / <str var> / **USING**"<num format>" | <num> / <num var> / <num expr> / **USING**"<str format>" | <str> / <str var> |

<num format> = "[–] [+] [0] <num1>.<num2>"

<str format> = "[–] <num3>"

Remarks: The optional , (comma) between <option>s causes a single space to appear in the print output. If it is omitted, the output is concatenated.

The optional ; (semicolon) allows subsequent **LPRINT**s to print on the same line. Using just **LPRINT** will print a blank line.

If **AT** is omitted, printing will start at the beginning of the line. If **AT** is used, this specifies at which column to start printing.

The keyword **USING** is a field formatter. All output is right-justified by default; using the optional – sign will cause output to be left-justified.

For <num format>, the optional + causes positive outputs to be signed. Output is normally padded with spaces, the optional 0 will pad the output with zeros. <num1> is the overall field width and <num2> is the number of decimal places.

For <str format>, <num3> is the overall field width.

Example: The following command statements cause the output shown beneath them to be printed. Explanations are given on page 2-30.

```
value1 = 123.456
hello$ = "hello"
world$ = "world"
column = 4
LPRINT "Start of program"
LPRINT USING "05.0" value1
LPRINT AT column; USING "10.2" value1
LPRINT AT column; USING "+10.2" value1
LPRINT USING "10" hello$
LPRINT USING "-10" hello$;
LPRINT world$
LPRINT hello$, world$
LPRINT hello$ world$
```

The output from the above program segment is:-

```
Start of program
00123
   123.46
   +123.46
     hello
hello     world
hello world
helloworld
```

The command statement `LPRINT "Start of program"` prints the string `Start of program` to the results store.

The statement `LPRINT USING "05.0" value1` prints the string `00123` (i.e. the value of the numeric variable `value1` with no decimal places, right-justified in a field width of 5, padded with zeros).

The statement `LPRINT AT column; USING "10.2" value1` prints `123.46` (i.e. the value of the numeric variable `value1` to two decimal places, starting at column 4 in a field width of 10).

The statement `LPRINT AT column; USING "+10.2" value1` prints `+123.46` (i.e. the *signed* value of the numeric variable `value1` to two decimal places, starting at column 4 in a field width of 10).

The statement `LPRINT USING "10" hello$` prints `hello` (the value of the string variable `hello$`) right-justified in a field width of 10.

The two statements `LPRINT USING "-10" hello$;` and `LPRINT world$` print `hello` left-justified in a field of 10, followed on the same line by `world`.

The statements `LPRINT hello$, world$` and `LPRINT hello$ world$` print `hello` and `world` on the same line with or without a space between them.

Function: To read back the current value of a measurement at any time during a program.

Syntax: **MEASURE** | AMDEPTH | <num var>
AFFREQ
AFLEVEL
FMDEVN
MODFREQ
RXDISTN
RXSINAD
RXSN
TXDISTN
TXFREQ
TXLEVEL
TXOFFSET
TXSINAD
TXSN

GSM, DCS and PCS only:
PKPHASE
PROFILE
RMSPHASE
RXBERI
RXBERII
RXFER
RXQUAL
RXRBERIB
RXRBERII
RXRSSI
TIMING

Remarks: <num var> is a numeric variable into which the measurement is returned. If <num var> = –1, the measurement is invalid. Ensure **RECEIVER AUTOTUNE** status is ON when **MEASURE TXFREQ** is used for analogue systems. Some digital measurement times are dependent on the number of samples specified in HELP SET-UP (see the Operating Manual, Chapter 3).

Measurement type	Units
AMDEPTH	%
AFFREQ	Hz
AFLEVEL	V
FMDEVN	Hz
MODFREQ	Hz
RXDISTN	%
RXSINAD	dB
RXSN	dB
TXDISTN	%
TXFREQ	Hz
TXLEVEL	W
TXOFFSET	Hz
TXSINAD	dB
TXSN	dB
PKPHASE	degrees
PROFILE	(1 = pass, 0 = fail)
RMSPHASE	degrees
RXBERI	%
RXBERII	%
RXFER	%
RXQUAL	none
RXRBERIB	%
RXRBERII	%
RXRSSI	none
TIMING	bits

In the following example, the AM depth is measured and returned in `fred`:-

```
MEASURE AMDEPTH fred
```

Function: To set the designated modulation generator's frequency.

Syntax: **MODGEN*n* FREQ** <num> | <num var> | <num expr> [Hz | kHz]

If the unit is omitted, the default unit is Hz.

Remarks: The *n* represents which generator is to be set, where *n* = 1, 2, or 3. The frequency is set using either a number, a numeric variable or a numeric expression. The unit is optional and is set using Hz or kHz. The frequency range for each generator is 10 Hz to 20 kHz.

This command is equivalent to using the MOD GEN, FREQ and DATA entry keys

Example: All of these examples assume that the modulation generators are ON. This can be done from MI-BASIC with the following command statement, where *n* = 1, 2 or 3:-

```
MODGENn STATUS ON
```

The level of the designated generator must be non-zero

The correct RF port must be selected.

1. This command statement uses a number for the frequency without specifying the optional units and sets the frequency of mod gen 1 to 100 Hz

```
MODGEN1 FREQ 100
```

2. This program segment sets the frequency of mod gen 2 to 5 kHz by using a numeric variable for the frequency and specifying the unit to be kHz.

```
value = 5
MODGEN2 FREQ value kHz
```

3. This program segment sets the frequency of mod generator 3 to 6 kHz by using a numeric expression for the frequency and specifying the unit to be kHz. It may be part of a loop where the frequency is incremented by 1 kHz each time the loop is passed through.

```
value = 5
MODGEN3 FREQ value + 1 kHz
```

Function: To set the designated modulation generator's level.

Syntax: **MODGEN*n* LEVEL** <num> | <num var> | <num expr> [Hz | kHz | % | RAD]

Hz and kHz are used if mod type is FM. If FM is selected, the default unit is Hz.
% is used if mod type is AM.
RAD is used if mod type is PM.

Remarks: The *n* represents which generator is to be set, where *n* = 1, 2 or 3. The level is set using either a number, a numeric variable or a numeric expression. The level must be non-zero. The correct mod type must be selected; if it is not, the level unit will be set to whatever the current mod type is.

This command is equivalent to using the MOD GEN, LEVEL and DATA entry keys.

Example: All of these examples assume that the mod generators are ON. This can be done from MI-BASIC with the following command statement, where *n* = 1, 2 or 3:-

```
MODGENn STATUS ON command.
```

These examples assume that the correct mod type has been selected. This can be done from MI-BASIC using one of the following command statements:-

```
MODTYPE FM
MODTYPE AM
MODTYPE PM
```

1. This command statement uses a number for the level and sets mod gen 1 level to 100 Hz.

   ```
   MODGEN1 LEVEL 100
   ```

2. This program segment uses a numeric variable for the level and sets mod gen 2 level to 500 Hz.

   ```
   value = 0.5
   MODGEN2 LEVEL value kHz
   ```

3. This program segment uses a numeric expression for the level and sets mod gen 3 level to 3.5 kHz. The command statement that contains the MODGEN command may be part of a loop where the level is incremented by 500 Hz each time the loop is passed through.

   ```
   value = 3
   MODGEN3 LEVEL value + 0.5 kHz
   ```

MODGEN*n* STATUS

Function: To switch the designated modulation generator ON or OFF.

Syntax: **MODGEN*n* STATUS** | ON
OFF |

Remarks: The *n* represents which generator is to be set, where *n* = 1, 2 or 3. This command is equivalent to using the *[gen n on OFF]* soft keys.

Example:

1. This command statement switches mod gen 1 ON:-

```
MODGEN1 STATUS ON
```

2. This command statement switches mod gen 2 OFF:-

```
MODGEN2 STATUS OFF
```

MODTYPE

Function: To set the type of modulation generator to FM (frequency modulation), AM (amplitude modulation) or PM (phase modulation).

Syntax: **MODTYPE** | AM
FM
PM |

Remarks: Only one type of modulation can be selected at one time, so the **MODTYPE** command statement affects the modulation sourced from mod gen 1, mod gen 2 and mod gen 3.

Example: This command statement sets the mod type to be FM:-

```
MODTYPE FM
```

NUMRESULTS

Function: To extract numerical results from the last test run.

Syntax: **NUMRESULTS** <num var1> [,...,<num var8>]

Remarks: There are 8 numeric fields. Numeric values that are returned in these fields are in the same units as those in which they were specified.

(See the **STRRESULTS** command for string results.)

The fields are in the following order:-

1. PASSED
2. ACTUAL
3. TARGET
4. LOWER
5. UPPER
6. AUX1
7. AUX2
8. AUX3

PASSED:	The state of the error flag. The error flag is 0 if passed and non-zero if the test failed. The failure codes are shown on page 2-36.
ACTUAL:	The actual reading
TARGET:	The target value
LOWER:	The lower limit
UPPER:	The upper limit
AUX 1, 2 & 3:	Auxiliary fields

The fields returned are dependent upon the previous test.

Example: In the following examples, the <num var>s that represent the fields returned by **NUMRESULTS** are given the same name as the field descriptions above; this does not have to be the case, the names of the fields can be anything as long as they are <num var>s.

1. In the following program segment, the AF frequency is tested. The required frequency is 3.4 kHz and the allowable error is 20%. **NUMRESULTS** is being used to extract the state of the error flag (did the test pass or fail?), the actual frequency (it should be 3.4 kHz ±20%), and the target frequency, which is 3.4 kHz. These results are then printed to the results store using the **LPRINT** command.

```
TEST AFFREQ REF 3.4KHZ,ERROR 20
NUMRESULTS error_flag,actual,target
LPRINT "Error flag is " error_flag
LPRINT "Actual is     " actual
LPRINT "Target is     " target
```

The result is:-

```
Error flag is 0
Actual is     3.75
Target is     3.4
```

2. In the following program segment, the AF level is tested. The reference level is 200 mV, and the error allowed is 10%. NUMRESULTS is used to extract the error flag and the target level; this means that a dummy field has to be used, as the fields that can be returned follow an order and only the first and third fields are required.

```
TEST AFLEVEL REF 200MV,ERROR 10
NUMRESULTS error_flag,dummy,target
LPRINT "Error flag is " error_flag
LPRINT "Target is     " target
```

The result is:-

```
Error flag is 0
Target is     200
```

NUMRESULTS error codes

The following list shows the codes that may be returned as the STATUS parameter when the NUMRESULTS command (or the PROG:NUMRESULTS? remote command - see Chapter 6) is used, together with their meanings. These codes will be generated following the execution of a TEST command (see page 2-53). The quoted error string will be returned if an STRRESULTS command (see page 2-52) is executed.

Code	Error String
0	"PASSED"
1	"*FAIL*"
101	"INVALID PARAMETERS"
102	"TIMED OUT"
103	"ABORTED BY USER"
104	"NOT ON CONTROL CHANNEL"
105	"NOT ON VOICE CHANNEL"
106	"NO RESPONSE"
107	"WRONG RESPONSE"
108	"NO SAT DETECTED"
109	"CHANNEL ASSIGNMENT"
110	"NO ALERT RESPONSE"
111	"NO ANSWER"
112	"SAT DROPOUT"
113	"ST DURATION"
114	"OUT OF RANGE"
115	"LOW MIC I/P SENS"
116	"NO MODULATION"
117	"NO DEMODULATION"
118	"UNSTABLE DEMOD"
119	"LOW DEMOD LEVEL"
120	"LOW SINAD LEVEL"
121	"CARRIER ON"
122	"CARRIER OFF"
123	"NO REGISTRATION"
124	"PRESSEL ON"
125	"PRESSEL OFF"
126	"NO PRESSEL ON"
127	"NO PRESSEL OFF"
128	"DROPPED OUT"
129	"NO SERIAL NO RESPONSE"
130	"CANNOT REGISTER"
131	"CALL REJECTED"
132	"RU UNAVAILABLE"
133	"CALL BACK"
134	"NO ACK ON TC"
135	"WRONG DISCONNECT QTY"
136	"RU CLEARED DOWN"
137	"UNEXPECTED MESSAGE"
138	"EMERGENCY CALL REJECTED"
139	"NO READING"
140	"NOT ON TRAFFIC CHANNEL"
141	"PROTOCOL ERROR"
142	"WRONG AUTHENTICATION"
143	"NO LOOPBACK"
144	"NO BURSTS"
145	"POOR SPEECH QUALITY"
146	"CALL ABORTED"
147	"NO RSSI REPORT"
148	"LOW SIGNAL QUALITY"
149	"FER TOO HIGH"
150	"NO AUTHENTICATON"
151	"MOBILE IN LOOPBACK"
152	"PASS/FAIL NOT SELECTED"
153	"RFGEN LEVEL RANGE ERROR"
154	"OPTION NOT FITTED"
155	"NOT ENABLED"
156	"NOT SUPPORTED"

PRINT AT

Function: To print to the screen at specific coordinates.

Syntax: **PRINT AT** | <column> | , | <row> | ; <option>
| <num var1> | | <num var2> |

<option> = | <num> |
| <num var> |
| <num exp> |
| <str> |
| <str var> |
| **USING**"<num format>" | <num> |
| <num var> |
| <num expr> |
| **USING**"<str format>" | <str> |
| <str var> |

<num format> = "[–] [+] [0] <num1>.<num2>"

<str format> = "[–] <num3>"

Remarks: The coordinates are specified as column, row. There are 58 columns numbered from –6 to 51, and 30 rows, 0 to 29.

The keyword **USING** is used as a field formatter. All output is normally right-justified; using the optional – sign will cause output to be left-justified. For numeric format, the optional + causes positive outputs to be signed. Output is normally padded with spaces, the optional 0 will pad the output with zeros. <num1> is the overall field width and <num2> is the number of decimal places.

For string format, <num3> is the field width.

Example: This program segment uses the **PRINT AT** command to demonstrate that characters can be printed anywhere on the screen:-

```
CLS
PRINT AT 10,10; "YOU CA    E SCREEN"
PRINT AT 15,11; "N    H"
PRINT AT 20,12; "T NO TN"
PRINT AT 15,13; "P          I"
PRINT AT 15,14; "RINT A     O"
PRINT AT 20,15; "T ANY P"
```

This program segment uses the **PRINT AT USING** command to demonstrate the field formatting.

```
val1 = 123.456
val2 = 678.90
ten = 10
CLS
PRINT AT ten, 10; USING "4.1" val1
PRINT AT ten, 11; USING "4.2" val1 + val2
```

The output from the first program is:-

```
YOU CA     E SCREEN
     N     H
           T NO TN
     P           I
     RINT A      O
           T ANY P
```

The output from the second program is:-

```
123.5
802.36
```

PRINTF AT

Function: To print flashing characters to the screen at specific coordinates.

Syntax: **PRINTF AT** | <num1> / <num var1> | , | <num 2> / <num var2> | ; <option>

<option> = | <num> / <num var> / <num exp> / <str> / <str var> / **USING**"<num format>" | <num> / <num var> / <num expr> | / **USING**"<str format>" | <str> / <str var> |

<num format> = "[–] [+] [0] <num1>.<num2>"

<str format> = "[–] <num3>"

Remarks: This command prints flashing text, otherwise it is exactly the same as the **PRINT AT** command.

Function: To disable printing from tests to the results store.

Syntax: **PRINTOFF**

Remarks: This command allows the user to customise the tests by formatting the results layout using the **LPRINT** command.

Example: In the following program segment, **PRINTOFF** stops the test printing the results to the results store.

The register status flag is read; if it is 1, **TEST REGISTER** is executed. If **TEST REGISTER** passes, some string results are printed into the results store using the **LPRINT** command. If the test fails, a fail message is printed into the results store using the **LPRINT** command.

In both cases (pass or fail) the format of the results is determined by **LPRINT** commands and not by the test itself.

```
PRINTOFF
handoff_count = 0
fail_count = 0

REM register and output mobile information
READPARAM REGISTER STATUS flag
IF flag = 1 THEN
      TEST REGISTER
REM if passed output mobile information
REM else put up error message
      NUMRESULTS error_flag
      IF error_flag = 0 THEN
            STRRESULTS dummy$, min$, dummy$, esn$
            LPRINT min$
            LPRINT
            LPRINT esn$
      ELSE
            LPRINT "**** REGISTRATION FAILED ****"
            fail_count = fail_count + 1
      ENDIF
      GETPAUSE
ENDIF
```

Function: To enable printing from tests to the results store.

Syntax: **PRINTON**

Remarks: This command enables the default printing of results.

Example: In the following program segment, **PRINTON** re-enables the printing of the results to the results store following a **PRINTOFF** command. The format of the results before the **PRINTON** command is determined by **LPRINT** command statements and not by the test itself. **TEST PLACECALL** is executed (the results from this test will be printed to the screen by the test, not the user) and the program continues.

```
PRINTOFF
handoff_count = 0
fail_count = 0

REM register and output mobile information
READPARAM REGISTER STATUS flag
IF flag = 1 THEN
   TEST REGISTER
REM if passed output mobile information
REM else put up error message
   NUMRESULTS error_flag
   IF error_flag = 0 THEN
       STRRESULTS dummy$, min$, dummy$, esn$
       LPRINT min$
       LPRINT
       LPRINT esn$
   ELSE
       LPRINT "**** REGISTRATION FAILED ****"
       fail_count = fail_count + 1
       ENDIF
       GETPAUSE
   ENDIF
   GETUSER user$
   GETDATE date$
   LPRINT
   LPRINT "TESTER:   ";user$
   LPRINT
   LPRINT "DATE:     ";date$
   LPRINT
   LPRINT
ENDIF

REM do initial call processing tests
PRINTON
TEST PLACECALL
NUMRESULTS error_flag
```

READPARAM

Function: To read back the setting of a parameter at any time during a test program.

Syntax: **READPARAM**

AFFREQ AFLEVEL FMDEVN MODFREQ RXDISTN RXEXPAND RXSENS RXSINAD RXSN TXCOMPRESS TXDISTN TXFREQ TXLEVEL TXLIMIT TXMODSENS TXNOISE TXSINAD TXSN CONFIGURATION DATADEVN DATAPERFORM DSATDEVN DTMFDECODE HANDOFF HOOKFLASH LANDCLEAR MOBILECLEAR PAGEMOBILE PKPHASE PLACECALL POWERLEVEL PROFILE PTTOFF PTTON REGISTER REGWARP RMSPHASE RXBERI RXBERII RXFER RXRBERIB RXRBERII RXRSSI SATDEVN SATFREQ SPEECHQUAL SPOTFREQ STDEVN STDURN STFREQ TIMING	<param> <num var>

Remarks: For the relevant parameters, see Chapter 3.

<num var> is the numeric variable into which the parameter value is returned.

Note: The READPARAM command can also be used in the form:-

```
READPARAM <str> <num var>
```

to read special parameters set via the WRITEPARAM command - see pages 2-42 and 2-57 for further information about reading and setting special parameters.

The following table shows the significance of the numerical values returned. For example, a numerical value of 2 for RXFILTER or TXFILTER indicates that the 5 kHz low-pass filter is selected. See also the example given below.

Parameter read	Status	Numeric value
STATUS, USEPRESSEL, RETRYENABLED	OFF ON	0 1
ACCPORT	LOGIC0 LOGIC1 LOGIC2 LOGIC3	0 1 2 3
RXFILTER, TXFILTER	NONE LP20KHZ LP5KHZ LP300HZ STDBP CCITT CMESS	0 1 2 3 4 5 6
CHANTYPE	WIDE NARROW ABOVE BELOW ROTATE LAST	0 1 2 3 4 5
MEASTYPE	WORSTCASE AVERAGE	0 1
WARPTYPE	WARPHILO WARPHIGH WARPLOW	0 1 2

Example: In this command statement, the status of **TXLIMIT** is read. **STATUS** is the parameter and `test_on` is the <num var> into which **TXLIMIT** status is returned. If the **STATUS** is ON, 1 is returned. If it is OFF, 0 is returned.

```
READPARAM TXLIMIT STATUS test_on
```

Reading special test parameters

As with the WRITEPARAM command (page 2-57), certain special parameters require a modified form of the READPARAM command, that is:-

READPARAM <str> <num var>

You can use this form of the command to read any of the special parameters used with the WRITEPARAM command, as listed on page 2-57, and also those shown below:-

"FIRST_VC" First voice channel†

"LAST_VC" Last voice channel†

"HANDOFF_INC" Handoff increment†

"FIRST_SLOT" First timeslot [GSM, DCS, PCS only]†

"LAST_SLOT" Last timeslot [GSM, DCS, PCS only]†

"SLOT_INC"	Slot increment [GSM, DCS, PCS only]†
"PCHAN"	Present channel
"PTYPE"	Present channel type [AMPS, TACS only]
"TIMESLOT"	Present timeslot [GSM, DCS, PCS only]

† As set on the SYSTEM PARAMETERS menu.

RECEIVER AUTOTUNE

Function: To switch the receiver's auto-tune on or off.

Syntax: **RECEIVER AUTOTUNE** | ON / OFF |

Example: This command statement switches the receiver's auto-tune off:-

```
RECEIVER AUTOTUNE OFF
```

RECEIVER CHAN

Function: To set the receiver's frequency to that of a specific channel.

This only applies to the frequency plan accessed using [HELP SET-UP], *[SET-UP]*, *[TEST OPTIONS]*, *[freq plan]*, and not a cellular/trunking plan. Hence this command would normally be used when writing programs for the PMR system only.

Syntax: **RECEIVER CHAN** | <num> / <num var> / <num expr> |

Remarks: The channel number can be in the range 1 - 9999.

Example: The following program segment sets the Tx frequency to that of channel 9:-

```
channel = 9
RECEIVER CHAN channel
```

RECEIVER FREQ

Function: To set the frequency to which the receiver is tuned.

Syntax: **RECEIVER FREQ** | <num> / <num var> / <num expr> | [Hz / kHz / MHz]

If the unit is omitted, the default unit is Hz.

Example: In the following example, the first line sets the frequency to which the receiver is tuned to the numeric field called `centre_freq`. The second line prints the result.

```
RECEIVER FREQ centre_freq
LPRINT centre_freq
```

REM

Function: To comment a program.

Syntax: **REM** <comments to end of line>

Remarks: This keyword enables commenting of the MI-BASIC program. Anything after the **REM** keyword up to the end of the line is ignored by the program.

Example: The **REM** command statement comments the program and is ignored:-

```
REM This is a comment.
```

RETURN

Function: To return from a subroutine.

Syntax: **RETURN**

Remarks:

Example: This program has a subroutine to print a message on the screen. The program goes around the loop 10 times, each time it jumps to the **LABEL** message, executes the subroutine, and returns from the subroutine when the **RETURN** keyword is executed. After 10 loops, the program terminates.

The subroutine is positioned after the **END** command.

```
        count = 0
LABEL start
        count = count + 1
        GOSUB message

        IF count < 10 THEN
              GOTO start
        ENDIF

        END

LABEL message
        CLRSTORE
        LPRINT "Hello"
        LPRINT "Count = ",USING "2.0" count
        RETURN
```

RFGEN CHAN

Function: To set the RF generator frequency to that of a specific channel.

Syntax: **RFGEN CHAN** <num> | <num var> | <num expr>

Remarks: The channel number can be in the range 1 - 9999.

This only applies to the frequency plan accessed using [HELP SET-UP], *[SET-UP]*, *[TEST OPTIONS]*, *[freq plan]*, and not a cellular/trunking plan. Hence this command would normally be used when writing programs for the PMR system only.

Example: The following program segment sets the RF gen frequency to that of channel 9:-

```
channel = 9
RFGEN CHAN channel
```

RFGEN FREQ

Function: To set the RF generator's frequency.

Syntax: **RFGEN FREQ** <num> | <num var> | <num expr> [Hz | kHz | MHz]

If the unit is omitted, the default unit is Hz.

Remarks: The frequency is set using either a number, a numeric variable or a numeric expression. The unit is optional and is set using Hz, kHz or MHz. The frequency range for the generator is 90 kHz to 1150 MHz.

This command is equivalent to using the RF GEN, FREQ and DATA entry keys.

Example: All of these examples assume the RF generator is ON. This can be done from MI-BASIC using the following command statement:-

```
RFGEN STATUS ON
```

The level of the generator must be non-zero. The level the generator can be set to will depend on which RF port is selected.

1. This command statement uses a number for the frequency without specifying the optional units and sets the RF generator frequency to 90 kHz.

   ```
   RFGEN FREQ 90000
   ```

2. This program segment sets the RF generator frequency to 500 kHz by using a numeric variable for the frequency and specifying the unit to be kHz.

   ```
   value = 500
   RFGEN FREQ value kHz
   ```

3. This program segment sets the RF generator frequency to 1.5 MHz by using a numeric expression for the frequency and specifying the unit to be MHz. This command may be part of a loop where the frequency is incremented by 1 MHz each time the loop is passed through.

   ```
   value = 0.5
   RFGEN FREQ value + 1 MHz
   ```

RFGEN LEVEL

Function: To set the RF generator's level.

Syntax: **RFGEN LEVEL** | <num> / <num var> / <num expr> | [dBm]

The unit is optional and defaults to dBm.

Remarks: The level is set using either a number, a numeric variable or a numeric expression. The level unit is dBm and optional. The level the generator can be set to will depend on which RF port is selected.

This command is equivalent to using the RF GEN, LEVEL and DATA entry keys.

Example: All of these examples assume that the RF generator is ON; this can be done from MI-BASIC using the following command statement:-

```
RFGEN STATUS ON
```

1. This command statement uses a number to set the RF generator level to 13 dBm.

```
RFGEN LEVEL 13 dBm
```

2. This program segment uses a numeric variable to set the RF generator level to –2 dBm.

```
harry = -2
RFGEN LEVEL harry dBm
```

3. This program segment uses a numeric expression to set the RF generator level to –5 dBm.

```
harry = -2
tom = -3
RFGEN LEVEL harry + tom
```

RFGEN STATUS

Function: To switch the RF generator ON or OFF.

Syntax: **RFGEN STATUS** | ON / OFF |

Remarks: This command is equivalent to using the *[gen n on OFF]* soft key.

Example:

1. This command statement switches the RF generator ON:-

```
RFGEN STATUS ON
```

2. This command statement switches the RF generator OFF:-

```
RFGEN STATUS OFF
```

Function: To set up the RF port configuration (TNC / N-type connectors).

Syntax: **RFPORT** | N_DUPLEX
| TNC_DUPLEX
| N_OUT_TNC_IN_DUPLEX
| TNC_OUT_N_IN_DUPLEX |

Remarks: This command is equivalent to using the RF SELECT front panel key.

To set up the RF port for simplex operation, set the RF port to a 2-port duplex mode and ignore the other RF port.

Example: This command statement configures the RF port to output RF on the TNC socket and input RF on the N-type socket:-

```
RFPORT TNC_OUT_N_IN_DUPLEX
```

This command statement configures the RF port to output and input RF on the N-type socket:-

```
RFPORT N_DUPLEX
```

Function: To read data from the RS232 port.

Syntax: **RS232_IN** <num var>

Remarks: This command reads the ASCII equivalent of the character that is sent. If this value is printed to the screen, the decimal value will be printed.

If the buffer is empty, –1 is returned.

Example: The following program segment receives:-
`hello`, <carriage return>, <line feed>, <line feed>, `world`, <carriage return>, <line feed>.

`h` = 68_{16} = ASCII for `h` = 104_{10}
`e` = 65_{16} = ASCII for `e` = 101_{10}
`l` = $6C_{16}$ = ASCII for `l` = 108_{10}
`l` = $6C_{16}$ = ASCII for `l` = 108_{10}
`o` = $6F_{16}$ = ASCII for `o` = 111_{10}

<carriage return> = $0D_{16}$ = ASCII for <carriage return> = 13_{10}
<line feed> = $0A_{16}$ = ASCII for <line feed> = 10_{10}

`w` = 77_{16} = ASCII for `w` = 119_{10}
`o` = $6F_{16}$ = ASCII for `o` = 111_{10}
`r` = 72_{16} = ASCII for `r` = 114_{10}
`l` = $6C_{16}$ = ASCII for `l` = 108_{10}
`d` = 64_{16} = ASCII for `d` = 100_{10}

<carriage return> = $0D_{16}$ = ASCII for <carriage return> = 13_{10}
<line feed> = $0A_{16}$ = ASCII for <line feed> = 10_{10}

```
LABEL start
      RS232_IN fred
      LPRINT fred
      GETPAUSE
      GOTO start
```

The decimal value printed to the screen will be:-

```
104.000000
101.000000
108.000000
108.000000
111.000000
13.000000
10.000000
119.000000
111.000000
114.000000
108.000000
100.000000
13.000000
10.000000
```

RS232_OUT

Function: To send data to the RS232 port.

Syntax: **RS232_OUT** <num> | <num var> | <num expr> | <str>

Remarks: Depending on the RS232 port configuration, either 7 bit (0 - 127) or 8 bit (0 - 255) numbers or characters can be sent.

Example: The program segment below will send
A, <carriage return>, B, <carriage return>, `hello world`
to the RS232 port:-

```
RS232_OUT 65
fred = 13
RS232_OUT fred
tom = 1
RS232_OUT 65 + tom
RS232_OUT 13
RS232_OUT "hello world"
```

$65_{10} = 41_{16}$ = ASCII for A
$13_{10} = D_{16}$ = ASCII for <carriage return>
$66_{10} = 42_{16}$ = ASCII for B

RXFILTER

Function: To set the Rx filter.

Syntax: **RXFILTER** NONE | LP20KHZ | LP5KHZ | LP300HZ | STDBP | CCITT | CMESS

Remarks:

LP20KHZ	Low-pass 3 dB @ 20 kHz
LP5KHZ	Low-pass 3 dB @ 5 kHz
LP300HZ	Low-pass 3 dB @ 300 Hz
STDBP	Standard Band-Pass 300 Hz to 3.4 kHz
CCITT	Psophometric Weighting, European
CMESS	Psophometric Weighting, North American

Example: The following command statement sets the Rx filter to 300 Hz low-pass:-

```
RXFILTER LP300HZ
```

Function: To draw and label a soft key at defined positions on the screen.

Syntax: **SOFTKEY** <num> | <num var> , NORMAL | BRIGHT | DELETE | CLEAR [<str1> | <str1> <str2>]

Remarks: The <num> / <num var> is used to define the soft key position. See Fig. 2-3 and Fig. 2-4 on page 2-61 for key numbers and positions.

NORMAL	Normal intensity MI-BASIC soft key
BRIGHT	Bright intensity MI-BASIC soft key
CLEAR	Clears the soft key but not the function
DELETE	Clears the soft key and masks the original function so that another soft key can be drawn in the same position and used for the duration of the program

<str1> / <str1> <str2> provide the optional labelling of the soft key. Where one label is provided, it is centred both vertically and horizontally. Where two labels are provided, they are positioned one above the other and are centred horizontally.

For soft keys 0 - 7 and 32 - 39, the maximum string length is 7 characters, and for soft keys 16 - 21, the maximum string length is 8 characters.

Example: The following program segment deletes soft key 19 (*[next store]* soft key), clears soft key 20 (*[clear store]* soft key), then pauses.

When the program is paused, if soft key 20 is pressed the soft key re-appears and functions as it should - i.e. changes to the next store - the key has been cleared but the function is not masked. If soft key 19 is pressed, nothing happens - the key has been deleted and its original function masked. When the program that contains this program segment stops, the screen is redrawn, hence the original soft keys are restored.

```
LABEL start
      SOFTKEY 19, DELETE
      WAIT 1000
      SOFTKEY 20, CLEAR
      GETPAUSE
      GOTO start
```

The following program segment uses a numeric variable for the key position and demonstrates the use of the **NORMAL** and **BRIGHT** keywords.

```
LABEL start
      fred = 16
      SOFTKEY fred,NORMAL "Hello" "There"
      SOFTKEY 17, BRIGHT "Bright" "key"
      GOTO start
```

Function: To convert a numeric variable into a string variable.

Syntax: **SPRINT** <str var> [<option>] | <num> | <num var> | <num expr> |

<option> = **USING** <num format> | <num> | <num var> | <num expr> |

<num format> = "[–] [+] [0] <num1>.<num2>"

Remarks: The **ATOF** command does the opposite, it converts a string variable into a numeric variable.

The keyword **USING** is used as a field formatter. All output is normally right-justified; using the optional – sign will cause output to be left-justified.

For numeric format, the optional + causes positive outputs to be signed. Output is normally padded with spaces; the optional 0 will pad the output with zeros. <num1> is the overall field width and <num2> is the number of decimal places.

Example: The following program segment uses the **SPRINT** keyword with a number, a numeric variable and a numeric expression. It also uses the **USING** keyword to format the fields.

```
value1 = 12.345
value2 = 34.5678
SPRINT first$,98.6543
SPRINT second$ USING "4.2" 98.6543
SPRINT third$,value1
SPRINT fourth$ USING "4.2" value1
SPRINT fifth$ USING "4.2" value1 + value2

LPRINT "first  = " first$
LPRINT "second = " second$
LPRINT "third  = " third$
LPRINT "fourth = " fourth$
LPRINT "fifth  = " fifth$
```

The output from this program segment is:-

```
first  =  98.654300
second = 98.65
third  =  12.345000
fourth = 12.35
fifth  = 46.91
```

Function: To read string results from the previous test.

Syntax: **STRRESULTS** <str var1> [,...,<str var8>]

Remarks: All tests produce some infomation which is stored in strings. There are 8 string fields; the first is the title of the test, the rest depend on which test has been executed.

(See the **NUMRESULTS** command for numerical results.)

The fields are in the following order:-

1. TITLE
2. STATUS
3. COMMENT1
4. COMMENT2
5. AUX1
6. AUX2
7. AUX3
8. AUX4

TITLE:	Test title
STATUS:	PASSED or *FAIL*
COMMENT:	Pass or fail summary report
AUX 1, 2, 3 & 4	Additional test information (test-dependent).

Example: In this example, the string variables that represent the fields returned by **STRRESULTS** are given the same name as the field descriptions above; this does not have to be the case, the names of the fields can be anything as long as they are string variables.

In the following program segment, the AF input has been set to 4 kHz. The required frequency is 3.4 kHz and the allowable error is 20%. **STRRESULTS** is used to read the string information in the three fields. These results are then printed to the results store using the **LPRINT** command.

```
TEST AFFREQ REF 3.4,ERROR 20
STRRESULTS title$,status$,comment1$
LPRINT "Title is " title$
LPRINT "Status is " status$
LPRINT "Comment 1 is " comment$
```

The output from this program is:-

```
AF FREQUENCY          PASSED 4.000 kHz +17.6%
Title is AF FREQUENCY
Status is PASSED
Comment 1 is 4.000kHz +17.6%
```

Function: To execute a pre-defined test.

Syntax: **TEST**

AFFREQ AFLEVEL FMDEVN MODFREQ RXDISTN RXEXPAND RXSENS RXSINAD RXSN TXCOMPRESS TXDISTN TXFREQ TXLEVEL TXLIMIT TXMODSENS TXNOISE TXSINAD TXSN	[<params>]
DATADEVN DATAPERFORM DSATDEVN DTMFDECODE ENDTCH HANDOFF HOOKFLASH LANDCLEAR LOOPBK MOBILECLEAR MOBLEV PAGEMOBILE PKPHASE PLACECALL POWERLEVEL PROFILE PTTOFF PTTON REGISTER REGWARP RMSPHASE RXBERI RXBERII RXFER RXRBERIB RXRBERII RXRSSI SATDEVN SATFREQ SPEECHQUAL STARTTCH STDEVN STDURN STFREQ TIMADV TIMING	

Remarks: The parameters for each test will depend on which system is used. For the relevant parameters, refer to Chapter 3.

The order that the parameters are entered in is not important. Where parameters are not specified, the default parameters are used. The default parameters can be changed using the **WRITEPARAM** command.

Note: Certain tests, such as HANDOFF and POWERLEVEL, make use of special parameters, such as VCHAN and VMAC, which have to be set prior to the test by using the **WRITEPARAM** command, as shown in the examples below. Where this is the case, details are given in the description of the test parameters in Chapter 3.

Examples: In the first example, the **RXSENS** test is executed. The modulation level, Rx filter and upper limit are specified. The other parameters for this test, i.e. status, averages and reference SINAD, are not specified, so the default values for these parameters are used.

```
TEST RXSENS MODLEVEL 5KHZ, RXFILTER NONE, UPPER -116dBm
```

In the following examples, the transmit power level of an AMPS or TACS mobile is set to power level 5, and the transmit power level of an NMT mobile is set to power level 3.

```
WRITEPARAM "VMAC" 5
TEST POWERLEVEL

WRITEPARAM "TRAFFIC_POWER" 3
TEST POWERLEVEL
```

In the following example, a handoff is performed to channel 100:-

```
WRITEPARAM "VCHAN" 100
TEST HANDOFF
```

TXFILTER

Function: To set the Tx filter.

Syntax: **TXFILTER** NONE | LP5KHZ | LP20KHZ | LP300HZ | STDBP | CCITT | CMESS

Remarks:

LP5KHZ	Low pass 3 dB @ 5 kHz
LP20KHZ	Low pass 3 dB @ 20 kHz
LP300HZ	Low pass 3 dB @ 300 Hz
STDBP	Standard Band Pass 300 Hz to 3.4 kHz
CCITT	Psophometric Weighting, European
CMESS	Psophometric Weighting, North American.

Example: The following command statement sets the Tx filter to 5 kHz low pass:-

```
TXFILTER LP5KHZ
```

WAIT

Function: To wait the specified number of milliseconds before continuing the program.

Syntax: **WAIT** <num> | <num var> | <num expr>

Remarks: The number of milliseconds to wait can be entered as a number, a numeric variable or a numeric expression.

Example: In the following program segment, a number, a numeric variable and a numeric expression are used to specify the number of milliseconds to wait:-

```
LABEL start
      CLRSTORE
      LPRINT "Waiting 1 second"
      WAIT 1000
      LPRINT "Waiting 2 seconds"
      delay = 2000
      WAIT delay
      LPRINT "Waiting 3 seconds"
      onesec = 1000
      WAIT delay + onesec
      GOTO start
```

WRITEPARAM

Function: To set a test parameter at any time.

Syntax:

WRITEPARAM	AFFREQ AFLEVEL FMDEVN MODFREQ RXDISTN RXEXPAND RXSENS RXSINAD RXSN TXCOMPRESS TXDISTN TXFREQ TXLEVEL TXLIMIT TXMODSENS TXNOISE TXSINAD TXSN	<params>
	CONFIGURATION DATADEVN DATAPERFORM DSATDEVN DTMFDECODE HANDOFF HOOKFLASH LANDCLEAR MOBILECLEAR PAGEMOBILE PKPHASE PLACECALL POWERLEVEL PROFILE PTTOFF PTTON REGISTER REGWARP RMSPHASE RXBERI RXBERII RXFER RXRBERIB RXRBERII RXRSSI SATDEVN SATFREQ SPEECHQUAL SPOTFREQ STDEVN STDURN STFREQ TIMING	

Remarks: For the relevant test parameters, and how they are used, see Chapter 3.

Example: In this command statement, the AF frequency reference is set to 2 kHz:-

```
WRITEPARAM AFFREQ REF 2KHZ
```

WRITEPARAM (continued)

Setting special test parameters

You can use the **WRITEPARAM** command to set certain special parameters by using the syntax shown below:-

WRITEPARAM <str> | <num> | <num var> | <num expr> |

where the <str> values are:-

"TRAFFIC_POWER"	Sets the mobile power level for **TEST POWERLEVEL** [NMT only].
"VMAC"	Sets the mobile power level for **TEST POWERLEVEL** [AMPS, TACS only].
"VCHAN"	Sets the channel for **TEST HANDOFF**.
"NEW_TIMESLOT"	Sets the timeslot for **TEST HANDOFF** [GSM, DCS and PCS only].
"NEW_LOOPBACK"	Sets the loopback mode for **TEST LOOPBK** [GSM, DCS and PCS only].
"NEW_MOBILELEV"	Sets the mobile power level for **TEST MOBLEV**, **TEST POWERLEVEL**, **TEST PROFILE** [GSM, DCS and PCS only].
"NEW_RF_POWER"	Sets the RF generator level for **TEST RXRSSI** [GSM, DCS and PCS only].
"NEW_TIME_ADV"	Sets the timing advance for **TEST TIMEADV, TEST TIMING** [GSM, DCS and PCS only].
"NEW_WARP_MODE"	Sets the warp mode for **TEST REGWARP** [GSM, DCS and PCS only].
	In this case, <num>, <num var> or <num exp> defines the offset as follows : - 0 = no carrier or data rate offset 1 = +25 parts/10^6 carrier and data rate offset 2 = –25 parts/10^6 carrier and data rate offset

Note: If you try to set a parameter value (e.g. mobile power level) that is outside the range of the mobile, this will be detected when the test is run and corrected accordingly.

Example: In this command statement, the mobile power level parameter is set to 10 for the following Power Level test:-

```
WRITEPARAM "NEW_MOBILELEV" 10
TEST POWERLEVEL
```

Instrument key decimal values

For use with **GETKEY** and **LASTKEY** commands.

Table 2-1 Key descriptions versus decimal values

Data keys 80 - 92, 96 - 100

0 key	80
1 key	81
2 key	82
3 key	83
4 key	84
5 key	85
6 key	86
7 key	87
8 key	88
9 key	89
Dec. point key	90
Minus key	91
Delete key	92
MHz, V	96
kHz, mV	97
Hz, µV	98
dB, %	99
dBm, rad	100

Function keys 64 - 73

Tx	64
RF GEN	65
Rx=Tx FREQ	66
MOD GEN	67
AF GEN	68
FREQ	69
LEVEL	70
DEVN DEPTH	71
INC	72
CHANNEL	73

Increment keys 76 - 79

ΔFREQ [⇩]	76
ΔFREQ [⇧]	77
δ [⇩]	78
δ [⇧]	79

Input - output select keys 112 - 114

EXT MOD INPUT, AC/DC	112
AF INPUT, AC/DC	113
RF TMC/N type, Routeing	114

Mode keys 48 - 55

Tx TEST	48
Rx TEST	49
DUPLEX TEST	50
TONES	51
SYSTEMS	52
RF TEST	53
AF TEST	54
HELP SET-UP	55

Print/store function keys 115 - 117

COPY	115
STORE	116
RECALL	117

Soft keys, bottom 16 - 21

Bottom left	16
Bottom right	21

Soft keys, left 32 - 39

Left top	32
Left bottom	39

Soft keys, right 0 - 7

Right top	0
Right bottom	7

Table 2-2 Decimal values versus key descriptions

0 - 7
Right soft keys

0	Top right
7	Bottom right

16 - 21
Bottom soft keys

16	Left bottom
21	Right bottom

32 - 39
Left soft keys

32	Top left
39	Bottom left

48 - 55
Mode keys

48	Tx TEST
49	Rx TEST
50	DUPLEX TEST
51	TONES
52	SYSTEMS
53	RF TEST
54	AF TEST
55	HELP SET-UP

64 - 73
Function keys

64	Tx
65	RF GEN
66	Rx=Tx FREQ
67	MOD GEN
68	AF GEN
69	FREQ
70	LEVEL
71	DEVN DEPTH
72	INC
73	CHANNEL

76 - 79
Increment keys

76	ΔFREQ [⇩]
77	ΔFREQ [⇧]
78	δ [⇩]
79	δ [⇧]

80 - 92, 96 - 100
Data keys

80	0 key
81	1 key
82	2 key
83	3 key
84	4 key
85	5 key
86	6 key
87	7 key
88	8 key
89	9 key
90	Decimal point key
91	Minus key
92	Delete key
96	MHz, V
97	kHz, mV
98	Hz, μV
99	dB, %
100	dBm, rad

112 - 114
Input - output select keys

112	EXT MOD INPUT, AC/DC
113	AF INPUT, AC/DC
114	RF TMC/N type, Routeing

115 - 117
Print/store function keys

115	COPY
116	STORE
117	RECALL

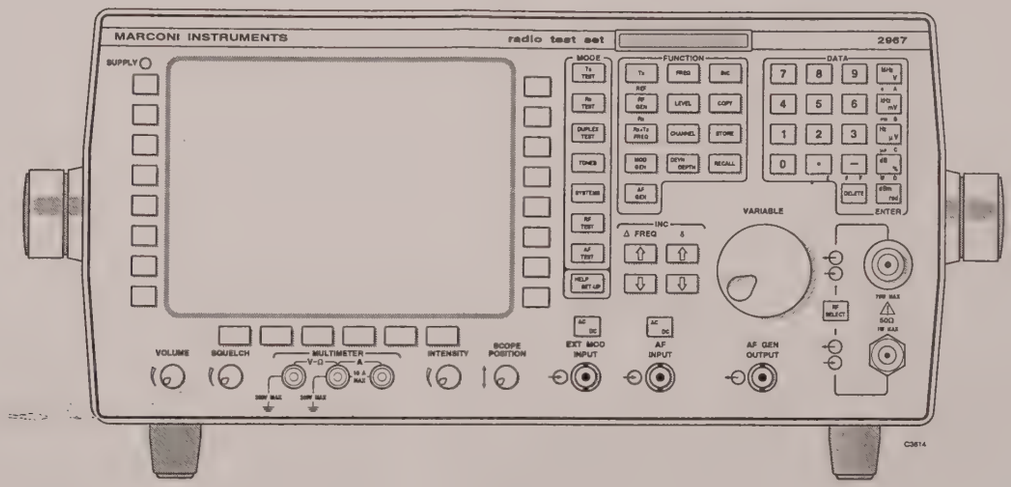


Fig. 2-3 Front panel layout

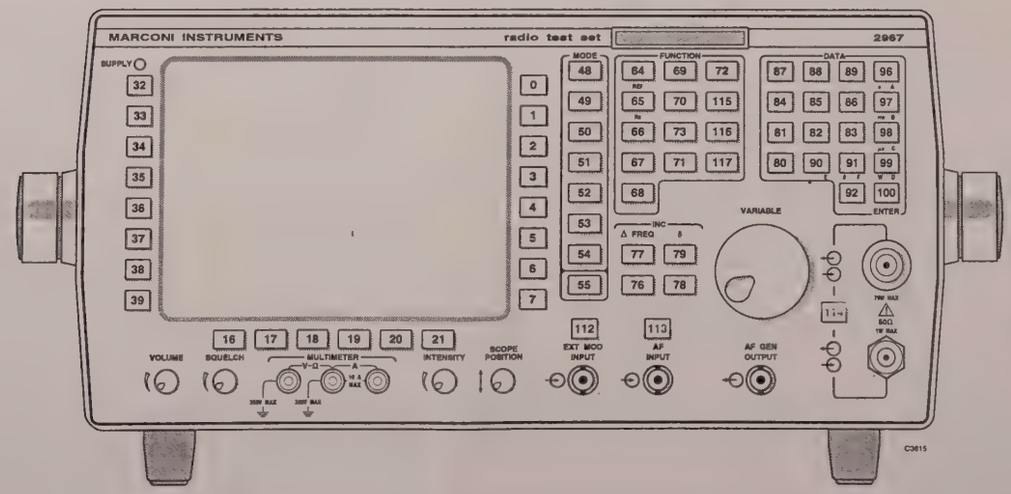


Fig. 2-4 Key code number definition

Front panel layout and key codes definition diagrams
(Figures 2-3 and 2-4)
overleaf

Chapter 3
TEST PARAMETERS

Contents

About this chapter 3-1
Setting test parameters 3-4
    Parameters used in automatic tests 3-4
    Setting special test parameters 3-4
Reading test parameters 3-4
    Reading special test parameters 3-4
Parametric test commands 3-5
Cellular and trunking test commands 3-21

List of tables

Table 3-1 Parametric tests 3-2
Table 3-2 Cellular and trunking tests 3-3

About this chapter

This chapter lists the **TEST** commands and gives details of the parameters associated with them. These parameters also apply to the **READPARAM** and **WRITEPARAM** commands if these commands are substituted for **TEST** at every occurrence. For syntax information, refer to Chapter 2.

Tables 3-1 and 3-2 serve as a contents page and show in which systems the **TEST/READPARAM/WRITEPARAM** commands are, or can be, used. Further information can be found in the Operating Manual Supplements for these systems, where their default parameters are listed. The Supplements are listed in *About this manual* at the front of this manual.

The specific system columns show which tests are included in that system's built-in test programs (✓), or are available for use in user-defined test programs for that system (≡).

Table 3-1 Parametric tests

COMMANDS	AMPS	TACS	MPT1327	NMT	PMR	GSM DCS PCS	See page
AFFREQ							3-5
AFLEVEL							3-5
FMDEVN							3-5
MODFREQ							3-6
RXDISTN	✓	✓	✓	✓	✓		3-7
RXEXPAND	✓	✓	✓	✓	✓		3-8
RXSENS	✓	✓	✓	✓	✓	✓	3-9
RXSINAD	✓	✓	✓	✓	✓		3-10
RXSN	✓	✓	✓	✓	✓		3-11
TXCOMPRESS	✓	✓	✓	✓	✓		3-12
TXDISTN	✓	✓	✓	✓	✓		3-13
TXFREQ	✓	✓	✓	✓	✓	✓	3-14
TXLEVEL			✓		✓		3-15
TXLIMIT	✓	✓	✓	✓	✓		3-16
TXMODSENS					✓		3-17
TXNOISE	✓	✓	✓	✓	✓		3-18
TXSINAD	✓	✓	✓	✓	✓		3-19
TXSN	✓	✓	✓	✓	✓		3-20

Table 3-2 Cellular and trunking tests

COMMANDS	AMPS	TACS	MPT1327	NMT	PMR	GSM DCS PCS	See page
CONFIGURATION	✓	✓	✓	✓	✓	✓	3-21
DATADEVN	✓	✓					3-21
DATAPERFORM	✓	✓					3-22
DSATDEVN	✓	✓					3-22
DTMFDECODE	✓	✓			✓		3-23
ENDTCH						✓	3-23
HANDOFF	✓	✓	✓	✓		✓	3-24
HOOKFLASH	✓	✓					3-25
LANDCLEAR	✓	✓	✓	✓		✓	3-25
LOOPBK						✓	3-26
MOBILECLEAR	✓	✓	✓	✓		✓	3-27
MOBLEV						✓	3-27
PAGEMOBILE	✓	✓	✓	✓		✓	3-28
PKPHASE						✓	3-29
PLACECALL	✓	✓	✓	✓		✓	3-30
POWERLEVEL	✓	✓		✓		✓	3-31
PROFILE						✓	3-33
PTTOFF			✓		✓		3-33
PTTON			✓		✓		3-34
REGISTER	✓	✓	✓	✓		✓	3-35
REGWARP						✓	3-36
RMSPHASE						✓	3-37
RXBERI						✓	3-37
RXBERII						✓	3-38
RXFER						✓	3-39
RXRBERIB						✓	3-40
RXRBERII						✓	3-41
RXRSSI						✓	3-42
SATDEVN	✓	✓		✓			3-43
SATFREQ	✓	✓		✓			3-43
SPEECHQUAL						✓	3-43
SPOTFREQ					✓		3-44
STARTTCH						✓	3-44
STDEVN	✓	✓					3-45
STDURN	✓	✓					3-45
STFREQ	✓	✓					3-46
TIMADV						✓	3-46
TIMING						✓	3-47

Setting test parameters

There are two standard methods for setting test parameters. You can set them:-

- Within a **TEST** command statement. For example:-

  ```
  TEST PLACECALL RFGENLEVEL -50DBM
  ```

  In this case, the **PLACECALL RFGENLEVEL** parameter is set *locally*, i.e. the -50 dBm value remains only for the duration of the test. If you want to set several parameters for a given test, you must put them all in a single command statement. The order is not important.

- Using a **WRITEPARAM** command statement. For example:-

  ```
  WRITEPARAM RXFER SAMPLES 200
  ```

  In this case, the **RXFER SAMPLES** parameter is set *globally*, i.e. the value you set becomes the default and appears on the AUTORUN PARAMETERS menu, and this new value will be used by subsequent RXFER tests unless the default is changed again.

Parameters used in automatic tests

Some parameters, such as **POWERLEVEL FIRSTVALUE** and **LASTVALUE**, or **TIMING FIRSTVALUE** and **LASTVALUE**, are used solely by automatic test programs (e.g. the Brief testing program), *not* by individual tests. So, although syntactically correct, command statements such as **TEST POWERLEVEL FIRSTVALUE 3** have no effect, as the **FIRSTVALUE** parameter is set locally (see above) and is not used by the **POWERLEVEL** test. You can, if you wish, use the **WRITEPARAM** command to set these parameters, in which case they will have the same effect in automatic test programs as if they had been set via the AUTORUN PARAMETERS menu. Similarly, you can use the **READPARAM** command to read the parameters.

Setting special test parameters

You can use the **WRITEPARAM** command to set certain special parameters - see the description of the **WRITEPARAM** command in Chapter 2 for further information.

Reading test parameters

The **READPARAM** command is used to read test parameters. The most common form of the command is:-

READPARAM <test_name> <param> <num var>

For example:-

```
READPARAM TXFREQ ERROR errval
```

The above command returns the transmitter frequency error in `errval`.

Some parameter values signify a particular selection. For example, for STATUS, 1 = ON, 0 = OFF. Other parameters in this category are ACCPORT, CHANTYPE, MEASTYPE, RETRYENABLED, RXFILTER, TXFILTER, USEPRESSEL and WARPTYPE. See the description of the **READPARAM** command in Chapter 2 for details.

Reading special test parameters

As with the **WRITEPARAM** command (see above), certain special parameters require a modified form of the **READPARAM** command - see the description of the **READPARAM** command in Chapter 2 for further information.

Parametric test commands

TEST AFFREQ

Test function : Tests AF frequency.

Function : AF frequency reference

Syntax : **TEST AFFREQ REF** <num> | <num var> | <num expr> [Hz | kHz | MHz]

Function : AF frequency error

Syntax : **TEST AFFREQ ERROR** <num> | <num var> | <num expr> %

TEST AFLEVEL

Test function : Tests AF level.

Function : AF level reference

Syntax : **TEST AFLEVEL REF** <num> | <num var> | <num expr> [VOLT | MVOLT]

Function : AF level error

Syntax : **TEST AFLEVEL ERROR** <num> | <num var> | <num expr> %

TEST FMDEVN

Test function : Tests FM deviation.

Function : FM deviation reference

Syntax : **TEST FMDEVN REF** <num> | <num var> | <num expr> [Hz | kHz | MHz]

Function : FM deviation error

Syntax : **TEST FMDEVN ERROR** <num> | <num var> | <num expr> %

TEST MODFREQ

Test function : Tests modulation frequency.

Function : Modulation frequency reference

Syntax : **TEST MODFREQ REF** <num> | <num var> | <num expr> [Hz | kHz | MHz]

Function : Modulation frequency error

Syntax : **TEST MODFREQ ERROR** <num> | <num var> | <num expr> %

TEST RXDISTN

Test function : Tests Rx distortion.

Function : Rx Distortion status

Syntax : **TEST RXDISTN STATUS** | ON
OFF |

Function : Rx Distortion averages

Syntax : **TEST RXDISTN AVERAGES** | <num>
<num var>
<num expr> |

Function : Rx Distortion RF generator level

Syntax : **TEST RXDISTN RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Rx Distortion modulation level

Syntax : **TEST RXDISTN MODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Rx Distortion Rx filter

Syntax : **TEST RXDISTN RXFILTER** | NONE
LP20KHZ
LP5KHZ
LP300HZ
STDBP
CCITT
CMESS |

Function : Rx Distortion upper limit

Syntax : **TEST RXDISTN UPPER** | <num>
<num var>
<num expr> | %

Function : Rx Distortion NB modulation level [NAMPS and NTACS only]

Syntax : **TEST RXDISTN NBMODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

TEST RXEXPAND

Test function : Tests Rx expansion.

Function : Rx Expansion status

Syntax : **TEST RXEXPAND STATUS** | ON
OFF |

Function : Rx Expansion averages

Syntax : **TEST RXEXPAND AVERAGES** | <num>
<num var>
<num expr> |

Function : Rx Expansion modulation level

Syntax : **TEST RXEXPAND MODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Rx Expansion Rx filter

Syntax : **TEST RXEXPAND RXFILTER** | NONE
LP20KHZ
LP5KHZ
LP300HZ
STDBP
CCITT
CMESS |

Function : Rx Expansion reference

Syntax : **TEST RXEXPAND REF** | <num>
<num var>
<num expr> |

Function : Rx Expansion error

Syntax : **TEST RXEXPAND ERROR** | <num>
<num var>
<num expr> | %

Function : Rx Expansion NB modulation level [NAMPS only]

Syntax : **TEST RXEXPAND NBMODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Rx Expansion modulation swing [PMR only]

Syntax : **TEST RXEXPAND SWING** | <num>
<num var>
<num expr> | DB

TEST RXSENS

Test function : Tests Rx sensitivity.

Function : Rx Sensitivity status

Syntax : **TEST RXSENS STATUS** | ON / OFF |

Function : Rx Sensitivity averages [Not GSM, DCS or PCS]

Syntax : **TEST RXSENS AVERAGES** | <num> / <num var> / <num expr> |

Function : Rx Sensitivity modulation level [Not GSM, DCS or PCS]

Syntax : **TEST RXSENS MODLEVEL** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

Function : Rx Sensitivity Rx filter [Not GSM, DCS or PCS]

Syntax : **TEST RXSENS RXFILTER** | NONE / LP20KHZ / LP5KHZ / LP300HZ / STDBP / CCITT / CMESS |

Function : Rx Sensitivity RF upper limit [For GSM this is for Class 4-5 mobiles only. For DCS/PCS this is for Class 1 and 2 mobiles only]

Syntax : **TEST RXSENS UPPER** | <num> / <num var> / <num expr> | DBM

Function : Rx Sensitivity reference SINAD [Not GSM, DCS or PCS]

Syntax : **TEST RXSENS REFSINAD** | <num> / <num var> / <num expr> | DB

Function : Rx Sensitivity NB modulation level [NAMPS only]

Syntax : **TEST RXSENS NBMODLEVEL** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

Function : Rx Sensitivity reference RBER [GSM, DCS and PCS only]

Syntax : **TEST RXSENS REFBER** | <num> / <num var> / <num expr> | %

Function : Rx Sensitivity number of samples [GSM, DCS and PCS only]

Syntax : **TEST RXSENS SAMPLES** | <num> / <num var> / <num expr> |

Function : Rx Sensitivity RF upper limit [For GSM Class 1-3 and DCS/PCS Class 3 mobiles]

Syntax : **TEST RXSENS HIUPPER** | <num> / <num var> / <num expr> | DBM

TEST RXSINAD

Test function : Tests Rx SINAD.

Function : Rx SINAD status

Syntax : **TEST RXSINAD STATUS** | ON
OFF |

Function : Rx SINAD averages

Syntax : **TEST RXSINAD AVERAGES** | <num>
<num var>
<num expr> |

Function : Rx SINAD RF level

Syntax : **TEST RXSINAD RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Rx SINAD modulation level

Syntax : **TEST RXSINAD MODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Rx SINAD Rx filter

Syntax : **TEST RXSINAD RXFILTER** | NONE
LP20KHZ
LP5KHZ
LP300HZ
STDBP
CCITT
CMESS |

Function : Rx SINAD lower limit

Syntax : **TEST RXSINAD LOWER** | <num>
<num var>
<num expr> | DB

Function : Rx SINAD NB modulation level [NAMPS and NTACS only]

Syntax : **TEST RXSINAD NBMODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

TEST RXSN

Test function : Tests Rx signal-to-noise ratio.

Function : Rx Signal-to-Noise ratio status

Syntax : **TEST RXSN STATUS** | ON / OFF |

Function : Rx Signal-to-Noise ratio averages

Syntax : **TEST RXSN AVERAGES** | <num> / <num var> / <num expr> |

Function : Rx Signal-to-Noise ratio RF level

Syntax : **TEST RXSN RFGENLEVEL** | <num> / <num var> / <num expr> | DBM

Function : Rx Signal-to-Noise ratio modulation level

Syntax : **TEST RXSN MODLEVEL** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

Function : Rx Signal-to-Noise ratio Rx filter

Syntax : **TEST RXSN RXFILTER** | NONE / LP20KHZ / LP5KHZ / LP300HZ / STDBP / CCITT / CMESS |

Function : Rx Signal-to-Noise ratio lower limit

Syntax : **TEST RXSN LOWER** | <num> / <num var> / <num expr> | DB

Function : Rx Signal-to-Noise ratio NB modulation level [NAMPS and NTACS only]

Syntax : **TEST RXSN NBMODLEVEL** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

TEST TXCOMPRESS

Test function : Tests Tx compression.

Function : Tx Compression status

Syntax : **TEST TXCOMPRESS STATUS** | ON
OFF |

Function : Tx Compression averages

Syntax : **TEST TXCOMPRESS AVERAGES** | <num>
<num var>
<num expr> |

Function : Tx Compression modulation level

Syntax : **TEST TXCOMPRESS MODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Tx Compression Tx filter

Syntax : **TEST TXCOMPRESS TXFILTER** | NONE
LP20KHZ
LP5KHZ
LP300HZ
STDBP
CCITT
CMESS |

Function : Tx Compression reference

Syntax : **TEST TXCOMPRESS REF** | <num>
<num var>
<num expr> |

Function : Tx Compression error

Syntax : **TEST TXCOMPRESS ERROR** | <num>
<num var>
<num expr> | %

Function : Tx Compression NB modulation level [NAMPS and NTACS only]

Syntax : **TEST TXCOMPRESS NBMODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Tx Compression modulation swing [PMR only]

Syntax : **TEST TXCOMPRESS SWING** | <num>
<num var>
<num expr> | DB

TEST TXDISTN

Test function : Tests Tx distortion.

Function : Tx Distortion status

Syntax : **TEST TXDISTN STATUS** | ON / OFF |

Function : Tx Distortion averages

Syntax : **TEST TXDISTN AVERAGES** | <num> / <num var> / <num expr> |

Function : Tx Distortion modulation level

Syntax : **TEST TXDISTN MODLEVEL** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

Function : Tx Distortion Tx filter

Syntax : **TEST TXDISTN TXFILTER** | NONE / LP20KHZ / LP5KHZ / LP300HZ / STDBP / CCITT / CMESS |

Function : Tx Distortion upper limit

Syntax : **TEST TXDISTN UPPER** | <num> / <num var> / <num expr> | %

Function : Tx Distortion NB modulation level [NAMPS and NTACS only]

Syntax : **TEST TXDISTN NBMODLEVEL** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

TEST TXFREQ

Test function : Tests Tx frequency against a specified reference.

Caution : For systems other than GSM, DCS or PCS, to set the reference frequency to the nominal frequency of the current channel you should use the GETTXFREQ command, as shown in the following example, before using the **TEST TXFREQ** command:-

```
GETTXFREQ tx_freq
tempfreq = tx_freq/1000000
```

You can then either set the **TXFREQ REF** parameter using the **WRITEPARAM** command before running the test, or directly via the **TEST TXFREQ** command, as shown below:-

```
WRITEPARAM TXFREQ REF tempfreq MHZ
TEST TXFREQ
```

or

```
TEST TXFREQ REF tempfreq MHZ
```

Function : Tx Frequency status

Syntax : **TEST TXFREQ STATUS** | ON / OFF |

Function : Tx Frequency measurement type [GSM, DCS and PCS only]

Syntax : **TEST TXFREQ MEASTYPE** | WORSTCASE / AVERAGE |

Function : Tx Frequency error tolerance

Syntax : **TEST TXFREQ ERROR** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

Function : Tx Frequency reference

Syntax : **TEST TXFREQ REF** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

Function : Tx Frequency NB error tolerance [NAMPS and NTACS only]

Syntax : **TEST TXFREQ NBERROR** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

Function : Tx Frequency error; number of bursts tested [GSM, DCS and PCS only]

Syntax : **TEST TXFREQ BURSTTST** | <num> / <num var> / <num expr> |

TEST TXLEVEL

Test function : Tests Tx level.

Function : Tx Level status

Syntax : **TEST TXLEVEL STATUS** [ON | OFF]

Function : Tx Level averages

Syntax : **TEST TXLEVEL AVERAGES** [<num> | <num var> | <num expr>]

Function : Tx Level upper limit

Syntax : **TEST TXLEVEL UPPER** [<num> | <num var> | <num expr>] [[WATT | MWATT]]

Function : Tx Level lower limit

Syntax : **TEST TXLEVEL LOWER** [<num> | <num var> | <num expr>] [[WATT | MWATT]]

TEST TXLIMIT

Test function : Tests Tx limiting.

Function : Tx Limiting status

Syntax : **TEST TXLIMIT STATUS** | ON
OFF |

Function : Tx Limiting averages

Syntax : **TEST TXLIMIT AVERAGES** | <num>
<num var>
<num expr> |

Function : Tx Limiting modulation level

Syntax : **TEST TXLIMIT MODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Tx Limiting Tx filter

Syntax : **TEST TXLIMIT TXFILTER** | NONE
LP20KHZ
LP5KHZ
LP300HZ
STDBP
CCITT
CMESS |

Function : Tx Limiting overload factor

Syntax : **TEST TXLIMIT OVERLOAD** | <num>
<num var>
<num expr> | DB

Function : Tx Limiting upper limit

Syntax : **TEST TXLIMIT UPPER** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Tx Limiting NB modulation level [NAMPS and NTACS only]

Syntax : **TEST TXLIMIT NBMODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Tx Limiting NB upper limit [NAMPS and NTACS only]

Syntax : **TEST TXLIMIT NBUPPER** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

TEST TXMODSENS

Test function : Tests Tx modulation sensitivity.

Function : Tx Modulation Sensitivity status

Syntax : **TEST TXMODSENS STATUS** [ON | OFF]

Function : Tx Modulation Sensitivity modulation level

Syntax : **TEST TXMODSENS MODLEVEL** <num> | <num var> | <num expr> [[Hz | kHz | MHz]]

Function : Tx Modulation Sensitivity Tx filter

Syntax : **TEST TXMODSENS TXFILTER** NONE | LP20KHZ | LP5KHZ | LP300HZ | STDBP | CCITT | CMESS

Function : Tx Modulation Sensitivity upper limit

Syntax : **TEST TXMODSENS UPPER** <num> | <num var> | <num expr> [[UVOLT | MVOLT | VOLT]]

Function : Tx Modulation Sensitivity lower limit

Syntax : **TEST TXMODSENS LOWER** <num> | <num var> | <num expr> [[UVOLT | MVOLT | VOLT]]

TEST TXNOISE

Test function : Tests Tx noise.

Function : Tx Noise status

Syntax : **TEST TXNOISE STATUS** | ON
OFF |

Function : Tx Noise averages

Syntax : **TEST TXNOISE AVERAGES** | <num>
<num var>
<num expr> |

Function : Tx Noise Tx filter

Syntax : **TEST TXNOISE TXFILTER** | NONE
LP20KHZ
LP5KHZ
LP300HZ
STDBP
CCITT
CMESS |

Function : Tx Noise upper limit

Syntax : **TEST TXNOISE UPPER** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

TEST TXSINAD

Test function : Tests Tx SINAD.

Function : Tx SINAD status

Syntax : **TEST TXSINAD STATUS** | ON
OFF |

Function : Tx SINAD averages

Syntax : **TEST TXSINAD AVERAGES** | <num>
<num var>
<num expr> |

Function : Tx SINAD modulation level

Syntax : **TEST TXSINAD MODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : Tx SINAD Tx filter

Syntax : **TEST TXSINAD TXFILTER** | NONE
LP20KHZ
LP5KHZ
LP300HZ
STDBP
CCITT
CMESS |

Function : Tx SINAD lower limit

Syntax : **TEST TXSINAD LOWER** | <num>
<num var>
<num expr> |

Function : Tx SINAD NB modulation level [NAMPS and NTACS only]

Syntax : **TEST TXSINAD NBMODLEVEL** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

TEST TXSN

Test function : Tests Tx signal-to-noise ratio.

Function : Tx Signal-to-Noise ratio status

Syntax : **TEST TXSN STATUS** { ON | OFF }

Function : Tx Signal-to-Noise ratio averages

Syntax : **TEST TXSN AVERAGES** { <num> | <num var> | <num expr> }

Function : Tx Signal-to-Noise ratio modulation level

Syntax : **TEST TXSN MODLEVEL** { <num> | <num var> | <num expr> } [{ Hz | kHz | MHz }]

Function : Tx Signal-to-Noise ratio Tx filter

Syntax : **TEST TXSN TXFILTER** { NONE | LP20KHZ | LP5KHZ | LP300HZ | STDBP | CCITT | CMESS }

Function : Tx Signal-to-Noise ratio lower limit

Syntax : **TEST TXSN LOWER** { <num> | <num var> | <num expr> } DB

Function : Tx Signal-to-Noise ratio NB modulation level [NAMPS and NTACS only]

Syntax : **TEST TXSN NBMODLEVEL** { <num> | <num var> | <num expr> } [{ Hz | kHz | MHz }]

Cellular and trunking test commands

READ/WRITEPARAM CONFIGURATION

Note : **CONFIGURATION** is *not* a test command; it can be used with the **READPARAM** or **WRITEPARAM** commands to read or set non-specific parameters.

Function : Switches duplex testing on or off (PMR only)

Syntax : **WRITEPARAM CONFIGURATION DUPLEX** | ON / OFF |

Function : Sets mobile power level for all tests except Power Level, Mobile Powerlevel and Power Profile [GSM, DCS and PCS only]

Syntax : **WRITEPARAM CONFIGURATION MOBILELEVEL** | <num> / <num var> / <num expr> |

Function : Sets timing advance for all tests except Timing and Timing Advance [GSM, DCS and PCS only]

Syntax : **WRITEPARAM CONFIGURATION TIME_ADVANCE** | <num> / <num var> / <num expr> | BITS

Function : Sets the Tx modulation type [All systems except GSM, DCS and PCS]

Syntax : **WRITEPARAM CONFIGURATION TXMODTYPE** | FFT / MODMETER |

Function : Sets the RF generator level for the following tests:- Power Level, Tx Frequency, Tx Peak Phase Error, Tx Power Profile, Tx RMS Phase Error, and Tx Timing [GSM, DCS and PCS only]

Syntax : **WRITEPARAM CONFIGURATION RFGENLEVEL** | <num> / <num var> / <num expr> | DBM

TEST DATADEVN

Test function : Tests data deviation.

Function : Data Deviation status

Syntax : **TEST DATADEVN STATUS** | ON / OFF |

Function : Data Deviation reference

Syntax : **TEST DATADEVN REF** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

Function : Data Deviation error tolerance

Syntax : **TEST DATADEVN ERROR** | <num> / <num var> / <num expr> | %

TEST DATAPERFORM

Test function : Tests data performance.

Function : Data Performance status

Syntax : **TEST DATAPERFORM STATUS** | ON
OFF |

Function : Data Performance RF generator level

Syntax : **TEST DATAPERFORM RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Data Performance lower limit

Syntax : **TEST DATAPERFORM LOWER** | <num>
<num var>
<num expr> | %

TEST DSATDEVN

Test function : Tests digital SAT deviation.

Function : DSAT Deviation status

Syntax : **TEST DSATDEVN STATUS** | ON
OFF |

Function : DSAT Deviation averages

Syntax : **TEST DSATDEVN AVERAGES** | <num>
<num var>
<num expr> |

Function : DSAT Deviation reference

Syntax : **TEST DSATDEVN REF** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : DSAT Deviation error tolerance

Syntax : **TEST DSATDEVN ERROR** | <num>
<num var>
<num expr> | %

TEST DTMFDECODE

Test function : Tests the mobile's ability to generate DTMF (Dual Tone Multiple Frequency) tones at the correct frequency.

Function : DTMF Decode status

Syntax : **TEST DTMFDECODE STATUS** | ON
OFF |

Function : DTMF Decode timeout

Syntax : **TEST DTMFDECODE TIMEOUT** | <num>
<num var>
<num expr> | SEC

Function : DTMF Decode use accessory port

Syntax : **TEST DTMFDECODE USEACCPORT** | ON
OFF |

Function : DTMF Decode set accessory port

Syntax : **TEST DTMFDECODE ACCPORT** | LOGIC0
LOGIC1
LOGIC2
LOGIC3 |

TEST ENDTCH

Test function : Ends the TCH Test mode and returns the test set to generating a control channel (BCCH). See also the TEST STARTTCH command (page 3-44).

TEST HANDOFF

Test function : Simulates the handoff of a call in progress from the current channel to the next channel.

Caution : To select the channel to hand off to, the following line needs to be inserted into the program before any **TEST HANDOFF** command:-

WRITEPARAM "VCHAN" | <num> / <num var> / <num expr> |

To set the timeslot in GSM, DCS and PCS, the following line also needs to be added:-

WRITEPARAM "NEW_TIMESLOT" | <num> / <num var> / <num expr> |

Alternatively, the **DEFAULT** command will initialise the channel and timeslot to the values set by the First Traffic Channel and Timeslot parameters on the SYSTEM PARAMETERS menu.

Example : The following program segment will do a handoff to channel 581:-

```
WRITEPARAM "VCHAN" 581
TEST HANDOFF
```

Note: *If the channel selected is not available, the instrument will find the next higher valid channel and hand off to that channel instead.*

To read back the channel that the mobile is on, the following line must be executed:-

```
READPARAM "VCHAN" <num var>
```

For GSM, DCS and PCS, to read back the timeslot that the mobile is currently on, the following line must be executed:-

```
READPARAM "TIMESLOT" <num var>
```

Function : Handoff status [GSM, DCS and PCS only]

Syntax : **TEST HANDOFF STATUS** | ON / OFF |

Function : Handoff RF generator level

Syntax : **TEST HANDOFF RFGENLEVEL** | <num> / <num var> / <num expr> | DBM

Function : Handoff timeout [MPT only]

Syntax : **TEST HANDOFF TIMEOUT** | <num> / <num var> / <num expr> | SEC

Function : Handoff channel type [NAMPS only]

Syntax : **TEST HANDOFF CHANTYPE** | WIDE / NARROW / ABOVE / BELOW / ROTATE / LAST |

TEST HANDOFF (continued)

Function : Handoff channel type [NTACS only]

Syntax : **TEST HANDOFF CHANTYPE** | WIDE
NARROW
ROTATE
LAST |

Function : Handoff mobile power level [NMT only]

Syntax : **TEST HANDOFF MOBILELEVEL** | <num>
<num var>
<num expr> |

TEST HOOKFLASH

Test function : Tests the ability of the mobile to send hook flash data.

Function : Hook Flash status

Syntax : **TEST HOOKFLASH STATUS** | ON
OFF |

Function : Hook Flash timeout

Syntax : **TEST HOOKFLASH TIMEOUT** | <num>
<num var>
<num expr> | SEC

Function : Hook Flash use accessory port

Syntax : **TEST HOOKFLASH USEACCPORT** | ON
OFF |

Function : Hook Flash set accessory port

Syntax : **TEST HOOKFLASH ACCPORT** | LOGIC0
LOGIC1
LOGIC2
LOGIC3 |

TEST LANDCLEAR

Test function : Simulates the termination of a call from a Land Station.

Function : Clear from Land status [GSM, DCS and PCS only]

Syntax : **TEST LANDCLEAR STATUS** | ON
OFF |

Function : Clear from Land RF generator level

Syntax : **TEST LANDCLEAR RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

TEST LOOPBK

Test function : Tests the ability of the mobile to enter loopback, with frame erasure (RBER), without frame erasure (BER), or to end loopback, while on a traffic channel. It also allows you to put the mobile into the appropriate mode for any measurements that follow.

Caution : To select the loopback mode to be used for the test, the following line needs to be inserted into the program before any **TEST LOOPBK** command:-

WRITEPARAM "NEW_LOOPBACK" | <num> | <num var> | <num expr> |

where <num>, <num var>, or <num exp> define the loopback mode as follows:-

0 = Loopback with frame erasure (RBER)
1 = Loopback without frame erasure (BER)
2 = No loopback

Alternatively, the **DEFAULT** command initialises the loopback mode to no loopback.

Example : The following program segment sets the mobile to loopback with frame erasure (RBER), measures the RBER Class II and FER, prints the results, and then ends loopback:-

```
WRITEPARAM "NEW_LOOPBACK" 0
TEST LOOPBK

MEASURE RXRBERII rber
MEASURE RXFER fer

LPRINT "Rx RBER CLASS II: " USING "2.3" rber; "%"
LPRINT "Rx FER          : " USING "2.3" fer; "%"

WRITEPARAM "NEW_LOOPBACK" 2
TEST LOOPBK
```

TEST MOBILECLEAR

Test function : Simulates the termination of a call from the mobile.

Function : Clear from Mobile status [GSM, DCS and PCS only]

Syntax : **TEST MOBILECLEAR STATUS** | ON / OFF |

Function : Clear From Mobile RF generator level

Syntax : **TEST MOBILECLEAR RFGENLEVEL** | <num> / <num var> / <num expr> | DBM

Function : Clear From Mobile timeout

Syntax : **TEST MOBILECLEAR TIMEOUT** | <num> / <num var> / <num expr> | SEC

Function : Clear From Mobile use accessory port

Syntax : **TEST MOBILECLEAR USEACCPORT** | ON / OFF |

Function : Clear From Mobile set accessory port

Syntax : **TEST MOBILECLEAR ACCPORT** | LOGIC0 / LOGIC1 / LOGIC2 / LOGIC3 |

TEST MOBLEV

Test function : Tests the ability of the mobile to set and report back a selected power level while on a traffic channel. It also allows you to change the mobile power level for any measurements that follow.

Caution : To select the mobile power level to be used for the test, the following line needs to be inserted into the program before any **TEST MOBLEV** command:-

WRITEPARAM "NEW_MOBILELEV" | <num> / <num var> / <num expr> |

Alternatively, the **DEFAULT** command initialises the mobile power level to level 9 for GSM or level 5 for DCS and PCS.

Example : The following program segment sets the mobile power level to 12, measures the power level, and prints the result:-

```
WRITEPARAM "NEW_MOBILELEV" 12
TEST MOBLEV

MEASURE TXLEVEL value
LPRINT "POWER: " USING "2.1" value; "dBm"
```

TEST PAGEMOBILE

Test function : Simulates the reception of a call from a Land Station.

Function : Page Mobile status [GSM, DCS and PCS only]

Syntax : **TEST PAGEMOBILE STATUS** | ON | OFF |

Function : Page Mobile RF generator level

Syntax : **TEST PAGEMOBILE RFGENLEVEL** | <num> | <num var> | <num expr> | DBM

Function : Page Mobile timeout

Syntax : **TEST PAGEMOBILE TIMEOUT** | <num> | <num var> | <num expr> | SEC

Function : Page Mobile use accessory port

Syntax : **TEST PAGEMOBILE USEACCPORT** | ON | OFF |

Function : Page Mobile set accessory port

Syntax : **TEST PAGEMOBILE ACCPORT** | LOGIC0 | LOGIC1 | LOGIC2 | LOGIC3 |

Function : Page Mobile channel type [NAMPS only]

Syntax : **TEST PAGEMOBILE CHANTYPE** | WIDE | NARROW | ABOVE | BELOW | ROTATE | LAST |

Function : Page Mobile channel type [NTACS only]

Syntax : **TEST PAGEMOBILE CHANTYPE** | WIDE | NARROW | ROTATE | LAST |

Function : Page Mobile mobile power level [NMT only]

Syntax : **TEST PAGEMOBILE MOBILELEVEL** | <num> | <num var> | <num expr> |

TEST PKPHASE

Test function : Tests the modulation accuracy of the mobile's transmitter (peak phase error).

Function : Peak Phase status

Syntax : **TEST PKPHASE STATUS** | ON
OFF |

Function : Peak Phase measurement type

Syntax : **TEST PKPHASE MEASTYPE** | WORSTCASE
AVERAGE |

Function : Peak Phase number of bursts tested

Syntax : **TEST PKPHASE BURSTTST** | <num>
<num var>
<num expr> |

Function : Peak Phase upper limit

Syntax : **TEST PKPHASE UPPER** | <num>
<num var>
<num expr> | DEG

TEST PLACECALL

Test function : Simulates the placing of a call from the mobile to a Land Station.

Function : Place Call status [GSM, DCS and PCS only]

Syntax : **TEST PLACECALL STATUS** | ON
OFF |

Function : Place Call RF generator level

Syntax : **TEST PLACECALL RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Place Call timeout

Syntax : **TEST PLACECALL TIMEOUT** | <num>
<num var>
<num expr> | SEC

Function : Place Call use accessory port

Syntax : **TEST PLACECALL USEACCPORT** | ON
OFF |

Function : Place Call set accessory port

Syntax : **TEST PLACECALL ACCPORT** | LOGIC0
LOGIC1
LOGIC2
LOGIC3 |

Function : Place Call channel type [NAMPS only]

Syntax : **TEST PLACECALL CHANTYPE** | WIDE
NARROW
ABOVE
BELOW
ROTATE
LAST |

Function : Place Call channel type [NTACS only]

Syntax : **TEST PLACECALL CHANTYPE** | WIDE
NARROW
ROTATE
LAST |

Function : Place Call mobile power level [NMT only]

Syntax : **TEST PLACECALL MOBILELEVEL** | <num>
<num var>
<num expr> | DBM

TEST POWERLEVEL

Test function : Tests the mobile transmitter power output level.

Caution : To select the mobile power level to change to and test, the following line needs to be inserted into the program before any **TEST POWERLEVEL** command:-

WRITEPARAM "VMAC" | <num> / <num var> / <num expr> | [for AMPS and TACS]

WRITEPARAM "TRAFFIC_POWER" | <num> / <num var> / <num expr> | [for NMT]

WRITEPARAM "NEW_MOBILELEV" | <num> / <num var> / <num expr> | [for GSM, DCS or PCS]

Note: For GSM, DCS or PCS, if the specified power level is invalid for the class of mobile, the test is run at the nearest valid power level. You can check the actual power level by using the **READPARAM "NEW_MOBILELEV" <num var>** command statement.

Example : The following program segment will test power level 4 [AMPS and TACS]:-

```
WRITEPARAM "VMAC" 4
TEST POWERLEVEL
```

The following program segment will test power level 2 [NMT]:-

```
WRITEPARAM "TRAFFIC_POWER" 2
TEST POWERLEVEL
```

Function : Power Level status

Syntax : **TEST POWERLEVEL STATUS** | ON / OFF |

Function : Power Level averages [not GSM, DCS or PCS]

Syntax : **TEST POWERLEVEL AVERAGES** | <num> / <num var> / <num expr> |

Note: The following parameters, **FIRSTVALUE**, **LASTVALUE**, and **INCREMENT**, have no effect if used in a **TEST POWERLEVEL** command statement (see page 3-4).

Function : First power level tested in a test program [GSM, DCS and PCS only]

Syntax : **TEST POWERLEVEL FIRSTVALUE** | <num> / <num var> / <num expr> |

Function : Last power level tested in a test program [GSM, DCS and PCS only]

Syntax : **TEST POWERLEVEL LASTVALUE** | <num> / <num var> / <num expr> |

Function : Power Level increment in a test program [GSM, DCS and PCS only]

Syntax : **TEST POWERLEVEL INCREMENT** | <num> / <num var> / <num expr> |

Function : Power Level upper limit (relative to reference power)

Syntax : **TEST POWERLEVEL RUPPER** | <num> / <num var> / <num expr> | DB

Function : Power Level lower limit (relative to reference power)

Syntax : **TEST POWERLEVEL RLOWER** | <num> / <num var> / <num expr> | DB

Function : Power Level high power upper limit (relative to reference power) for the highest power of which the mobile is capable and for power level 30 for PCS

Syntax : **TEST POWERLEVEL HIRUPPER** | <num> / <num var> / <num expr> | DB

Function : Power Level high power lower limit (relative to reference power) for the highest power of which the mobile is capable and for power level 30 for PCS

Syntax : **TEST POWERLEVEL HIRLOWER** | <num> / <num var> / <num expr> | DB

Function : Power Level mid power upper limit (relative to reference power) for mobile power levels 9-13 [DCS and PCS only]

Syntax : **TEST POWERLEVEL MIDRUPPER** | <num> / <num var> / <num expr> | DB

Function : Power Level mid power lower limit (relative to reference power) for mobile power levels 9-13 [DCS and PCS only]

Syntax : **TEST POWERLEVEL MIDRLOWER** | <num> / <num var> / <num expr> | DB

Function : Power Level low power upper limit (relative to reference power) for mobile power levels 16-19 [GSM] or 14-15 [DCS/PCS]

Syntax : **TEST POWERLEVEL LORUPPER** | <num> / <num var> / <num expr> | DB

Function : Power Level low power lower limit (relative to reference power) for mobile power levels 16-19 [GSM] or 14-15 [DCS/PCS]

Syntax : **TEST POWERLEVEL LORLOWER** | <num> / <num var> / <num expr> | DB

TEST PROFILE

Test function : Tests the conformance of the mobile's burst power profile to the GSM Specification.

Caution : To select the power level to change to and test, the following line needs to be inserted into the program before any **TEST PROFILE** command:-

WRITEPARAM "NEW_MOBILELEV" | <num> | <num var> | <num expr> |

Note: If the specified power level is invalid for the class of mobile, the test is run at the nearest valid power level. You can check the actual power level by using the **READPARAM "NEW_MOBILELEV" <num var>** command statement.

Example : The following program segment will test at power level 4:-

```
WRITEPARAM "NEW_MOBILELEV" 4
TEST PROFILE
```

Note: The following parameters, **FIRSTVALUE**, **LASTVALUE**, and **INCREMENT**, have no effect if used in a **TEST PROFILE** command statement (see page 3-4).

Function : Power Profile first power level tested in a test program

Syntax : **TEST PROFILE FIRSTVALUE** | <num> | <num var> | <num expr> |

Function : Power Profile last power level tested in a test program

Syntax : **TEST PROFILE LASTVALUE** | <num> | <num var> | <num expr> |

Function : Power Profile power level increment in a test program

Syntax : **TEST PROFILE INCREMENT** | <num> | <num var> | <num expr> |

TEST PTTOFF

Test function : Stops the mobile transmitting.

Function : PTT OFF timeout

Syntax : **TEST PTTOFF TIMEOUT** | <num> | <num var> | <num expr> | SEC

Function : PTT OFF use pressels [MPT and PMR only]

Syntax : **TEST PTTOFF USEPRESSEL** | ON | OFF |

TEST PTTON

Test function : Starts the mobile transmitting.

Function : PTT ON timeout

Syntax : **TEST PTTON TIMEOUT** | <num> | <num var> | <num expr> | SEC

Function : PTT ON use pressels [MPT and PMR only]

Syntax : **TEST PTTON USEPRESSEL** | ON | OFF |

Function : PTT ON use accessory port

Syntax : **TEST PTTON USEACCPORT** | ON | OFF |

Function : PTT ON set accessory port

Syntax : **TEST PTTON ACCPORT** | LOGIC0 | LOGIC1 | LOGIC2 | LOGIC3 |

TEST REGISTER

Test function : Tests the ability of the mobile to register.

Function : Registration status

Syntax : **TEST REGISTER STATUS** | ON
OFF |

Function : Registration authentication check [GSM only]

Syntax : **TEST REGISTER AUTHCHK** | ON
OFF |

Function : Registration RF generator level

Syntax : **TEST REGISTER RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Registration timeout [NMT, GSM, DCS and PCS only]

Syntax : **TEST REGISTER TIMEOUT** | <num>
<num var>
<num expr> | SEC

Function : Registration retry enabled [AMPS, TACS, GSM, DCS and PCS only]

Syntax : **TEST REGISTER RETRYENABLED** | ON
OFF |

Function : Registration set accessory port [NMT only]

Syntax : **TEST REGISTER ACCPORT** | LOGIC0
LOGIC1
LOGIC2
LOGIC3 |

Function : Registration use accessory port [NMT only]

Syntax : **TEST REGISTER USEACCPORT** | ON
OFF |

Function : Registration mobile power level [NMT only]

Syntax : **TEST REGISTER MOBILELEVEL** | <num>
<num var>
<num expr> |

TEST REGWARP

Test function : To test the ability of the mobile to register with the instrument carrier frequency and data rate offset by 25 parts/10^6 above or below the nominal carrier frequency and data rate.

Caution : To select the carrier and data rate offset, the following line needs to be inserted before any TEST REGWARP command : -

WRITEPARAM "NEW_WARP_MODE" | <num> / <num var> / <num exp> |

Where <num>, <num var> or <num exp> defines the offset as follows: -

0 = no carrier or data rate offset
1 = +25 parts/10^6 carrier and data rate offset
2 = –25 parts/10^6 carrier and data rate offset

The DEFAULT command will initialise to no offset.

Example : The following program segment reads the REGWARP WARPTYPE parameter (see below) and, depending on its value, performs a registration warp test with +25 parts/10^6 offset, or with –25 parts/10^6 offset, or with +25 parts/10^6 and –25 parts/10^6 offsets.

```
READPARAM REGWARP WARPTYPE warp_type
IF warp_type = 0 OR warp_type = 1 THEN
    WRITEPARAM "NEW_WARP_MODE" 1
    TEST REGWARP
ENDIF

IF warp_type = 0 OR warp_type = 2 THEN
    WRITEPARAM "NEW_WARP_MODE" 2
    TEST REGWARP
ENDIF
```

Function : Registration warp status

Syntax : **TEST REGWARP STATUS** | ON / OFF |

Note: The parameter **WARPTYPE** has no effect if used in a **TEST REGWARP** command statement. The example above shows a typical use for this parameter.

Function : Registration warp type

Syntax : **TEST REGWARP WARPTYPE** | WARPHILO / WARPHIGH / WARPLOW |

Function : Registration warp RF generator level

Syntax : **TEST REGWARP RFGENLEVEL** | <num> / <num var> / <num exp> | DBM

Function : Registration warp authentication check

Syntax : **TEST REGWARP AUTHCHK** | ON / OFF |

Function : Registration warp timeout

Syntax : **TEST REGWARP TIMEOUT** | <num> / <num var> / <num expr> | SEC

Function : Registration warp retry enabled

Syntax : **TEST REGWARP RETRYENABLED** | ON / OFF |

TEST RMSPHASE

Test function : Tests the modulation accuracy of the mobile's transmitter (RMS phase error).

Function : RMS Phase status

Syntax : **TEST RMSPHASE STATUS** | ON / OFF |

Function : RMS Phase measurement type

Syntax : **TEST RMSPHASE MEASTYPE** | WORSTCASE / AVERAGE |

Function : RMS Phase number of bursts tested

Syntax : **TEST RMSPHASE BURSTTST** | <num> / <num var> / <num expr> |

Function : RMS Phase upper limit

Syntax : **TEST RMSPHASE UPPER** | <num> / <num var> / <num expr> | DEG

TEST RXBERI

Test function : Tests the mobile receiver's bit error rate for Class I bits.

Function : Rx BER Class I status

Syntax : **TEST RXBERI STATUS** | ON / OFF |

Function : Rx BER Class I RF generator level for mobile power Classes 1 to 3 [GSM only] or Class 3 [DCS and PCS only]

Syntax : **TEST RXBERI HIRFGENLEVEL** | <num> / <num var> / <num expr> | DBM

Function : Rx BER Class I RF generator level for mobile power Classes 4 and 5 [GSM only] or Classes 1 and 2 [DCS and PCS only]

Syntax : **TEST RXBERI RFGENLEVEL** | <num> / <num var> / <num expr> | DBM

Function : Rx BER Class I number of samples tested

Syntax : **TEST RXBERI SAMPLES** | <num> / <num var> / <num expr> |

Function : Rx BER Class I upper limit

Syntax : **TEST RXBERI UPPER** | <num> / <num var> / <num expr> | %

TEST RXBERII

Test function : Tests the mobile receiver's bit error rate for Class II bits.

Function : Rx BER Class II status

Syntax : **TEST RXBERII STATUS** | ON / OFF |

Function : Rx BER Class II RF generator level for mobile power Classes 1 to 3 [GSM only] or Class 3 [DCS and PCS only]

Syntax : **TEST RXBERII HIRFGENLEVEL** | <num> / <num var> / <num expr> | DBM

Function : Rx BER Class II RF generator level for mobile power Classes 4 and 5 [GSM only] or Classes 1 and 2 [DCS and PCS only]

Syntax : **TEST RXBERII RFGENLEVEL** | <num> / <num var> / <num expr> | DBM

Function : Rx BER Class II number of samples tested

Syntax : **TEST RXBERII SAMPLES** | <num> / <num var> / <num expr> |

Function : Rx BER Class II upper limit

Syntax : **TEST RXBERII UPPER** | <num> / <num var> / <num expr> | %

TEST RXFER

Test function : Tests the mobile receiver's frame erasure rate.

Function : Rx Frame Erasure Rate status

Syntax : **TEST RXFER STATUS** | ON
OFF |

Function : Rx FER RF generator level for mobile power Classes 1 to 3 [GSM only] or Class 3 [DCS and PCS only]

Syntax : **TEST RXFER HIRFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Rx FER RF generator level for mobile power Classes 4 and 5 [GSM only] or Classes 1 and 2 [DCS and PCS only]

Syntax : **TEST RXFER RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Rx FER number of samples tested

Syntax : **TEST RXFER SAMPLES** | <num>
<num var>
<num expr> |

Function : Rx FER upper limit

Syntax : **TEST RXFER UPPER** | <num>
<num var>
<num expr> | %

TEST RXRBERIB

Test function : Tests the mobile receiver's residual bit error rate for Class Ib bits.

Function : Rx RBER Class Ib status

Syntax : **TEST RXRBERIB STATUS** | ON
OFF |

Function : Rx RBER Class Ib RF generator level for mobile power Classes 1 to 3 [GSM only] or Class 3 [DCS and PCS only]

Syntax : **TEST RXRBERIB HIRFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Rx RBER Class Ib RF generator level for mobile power Classes 4 and 5 [GSM only] or Classes 1 and 2 [DCS and PCS only]

Syntax : **TEST RXRBERIB RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Rx RBER Class Ib number of samples tested

Syntax : **TEST RXRBERIB SAMPLES** | <num>
<num var>
<num expr> |

Function : Rx RBER Class Ib upper limit

Syntax : **TEST RXRBERIB UPPER** | <num>
<num var>
<num expr> | %

TEST RXRBERII

Test function : Tests the mobile receiver's residual bit error rate for Class II bits.

Function : Rx RBER Class II status

Syntax : **TEST RXRBERII STATUS** | ON
OFF |

Function : Rx RBER Class II RF generator level for mobile power Classes 1 to 3 [GSM only] or Class 3 [DCS and PCS only]

Syntax : **TEST RXRBERII HIRFGENLEVE** | <num>
<num var>
<num expr> | DBM

Function : Rx RBER Class II RF generator level for mobile power Classes 4 and 5 [GSM only] or Classes 1 and 2 [DCS and PCS only]

Syntax : **TEST RXRBERII RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Rx RBER Class II number of samples tested

Syntax : **TEST RXRBERII SAMPLES** | <num>
<num var>
<num expr> |

Function : Rx RBER Class II upper limit

Syntax : **TEST RXRBERII UPPER** | <num>
<num var>
<num expr> | %

TEST RXRSSI

Test function : Tests the accuracy of the mobile's Received Signal Strength Indicator at a selected RF generator level

Caution : To select the RF generator level, the following line needs to be inserted into the program before any **TEST RXRSSI** command:-

WRITEPARAM "NEW_RF_POWER" | <num> / <num var> / <num expr> |

Alternatively, the **DEFAULT** command will initialise the level to –80 dBm.

Example : The following program segment will test the Rx RSSI report at –48.5 dBm:-

```
WRITEPARAM "NEW_RF_POWER" -48.5
TEST RXRSSI
```

Function : Rx RSSI status

Syntax : **TEST RXRSSI STATUS** | ON / OFF |

Note: The following parameters, **FIRSTVALUE**, **LASTVALUE**, and **DECREMENT**, have no effect if used in a **TEST RXRSSI** command statement (see page 3-4).

Function : Rx RSSI first RF generator level used in a test program

Syntax : **TEST RXRSSI FIRSTVALUE** | <num> / <num var> / <num expr> | DBM

Function : Rx RSSI last RF generator level used in a test program

Syntax : **TEST RXRSSI LASTVALUE** | <num> / <num var> / <num expr> | DBM

Function : Rx RSSI RF generator level decrement used in a test program

Syntax : **TEST RXRSSI DECREMENT** | <num> / <num var> / <num expr> | DB

Function : Rx RSSI error for RF generator levels >= –70 dBm

Syntax : **TEST RXRSSI HIERROR** | <num> / <num var> / <num expr> | DB

Function : Rx RSSI error for RF generator levels < –70 dBm

Syntax : **TEST RXRSSI ERROR** | <num> / <num var> / <num expr> | DB

TEST SATDEVN

Test function : Measures the SAT deviation.

Function : SAT Deviation status

Syntax : **TEST SATDEVN STATUS** | ON
OFF |

Function : SAT Deviation reference

Syntax : **TEST SATDEVN REF** | <num>
<num var>
<num expr> | [| Hz
kHz
MHz |]

Function : SAT Deviation error tolerance

Syntax : **TEST SATDEVN ERROR** | <num>
<num var>
<num expr> | %

TEST SATFREQ

Test function : Measures the SAT frequency.

Function : SAT Frequency status

Syntax : **TEST SATFREQ STATUS** | ON
OFF |

Function : SAT Frequency error tolerance

Syntax : **TEST SATFREQ ERROR** | <num>
<num var>
<num expr> | %

TEST SPEECHQUAL

Test function : Tests the quality of the complete audio/radio/audio path from the mobile's microphone/transmitter to the mobile's receiver/earpiece.

Function : Speech Quality status

Syntax : **TEST SPEECHQUAL STATUS** | ON
OFF |

Function : Speech Quality RF generator level

Syntax : **TEST SPEECHQUAL RFGENLEVEL** | <num>
<num var>
<num expr> | DBM

Function : Speech Quality timeout

Syntax : **TEST SPEECHQUAL TIMEOUT** | <num>
<num var>
<num expr> | SEC

TEST SPOTFREQ

Note : SPOTFREQ is *not* a test; it is a store for spot frequency parameters that is used with the **READPARAM** and **WRITEPARAM** commands (PMR only). It is included in this listing for completeness.

Function : Spot Frequency status

Syntax : **WRITEPARAM SPOTFREQ STATUS** | ON / OFF |

Function : Spot frequency for Tx channel *n*, where *n* = 1 to 16

Syntax : **WRITEPARAM SPOTFREQ TXCHAN*n*** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

Function : Spot frequency for Rx channel *n*, where *n* = 1 to 16

Syntax : **WRITEPARAM SPOTFREQ RXCHAN*n*** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

TEST STARTTCH

Test function : Puts the test set into TCH Test mode on a selected traffic channel. See also the TEST ENDTCH command (page 3-23).

Caution : To select the traffic channel that the test set will go to, the following line needs to be inserted into the program before any **TEST STARTTCH** command:-

WRITEPARAM "VCHAN" | <num> / <num var> / <num expr> |

Alternatively, the DEFAULT command will initialise the traffic channel to the value set by the First Traffic Channel parameter on the SYSTEM PARAMETERS menu.

Example : The following program segment puts the test set into TCH test mode on traffic channel 1, measures the power level, and prints the result. It then changes to channel 124, measures the power level, prints the result, and ends TCH Test mode.

```
WRITEPARAM "VCHAN" 1
TEST STARTTCH

MEASURE TXLEVEL value
LPRINT "POWER: " USING "2.1" value; "dBm"

REM Change channel using the HANDOFF command
WRITEPARAM "VCHAN" 124
TEST HANDOFF

MEASURE TXLEVEL value
LPRINT "POWER: " USING "2.1" value; "dBm"

TEST ENDTCH
```

TEST STDEVN

Test function : Tests the deviation of the mobile's signalling tone.

Function: ST Deviation status

Syntax: **TEST STDEVN STATUS** [ON | OFF]

Function: ST Deviation reference

Syntax: **TEST STDEVN REF** [<num> | <num var> | <num expr>] [[Hz | kHz | MHz]]

Function: ST Deviation error tolerance

Syntax: **TEST STDEVN ERROR** [<num> | <num var> | <num expr>] %

TEST STDURN

Test function : Tests the duration of the mobile's signalling tone.

Function: ST Duration status

Syntax: **TEST STDURN STATUS** [ON | OFF]

Function: ST Duration error tolerance

Syntax: **TEST STDURN ERROR** [<num> | <num var> | <num expr>] %

TEST STFREQ

Test function : Tests the frequency of the mobile's signalling tone..

Function: ST Frequency status

Syntax: **TEST STFREQ STATUS** | ON / OFF |

Function: ST Frequency error tolerance

Syntax: **TEST STFREQ ERROR** | <num> / <num var> / <num expr> | %

Function: ST Frequency reference

Syntax: **TEST STFREQ REF** | <num> / <num var> / <num expr> | [| Hz / kHz / MHz |]

TEST TIMADV

Test function : Tests the ability of the mobile to set and report back a selected timing advance while on a traffic channel. It also allows you to change the timing advance for any measurements that follow.

Caution : To select the timing advance to be set, the following line needs to be inserted into the program before any **TEST TIMADV** command:-

WRITEPARAM "NEW_TIME_ADV" | <num> / <num var> / <num expr> |

Alternatively, the DEFAULT command will initialise the timing advance to 0 bits.

Example : The following program segment sets the timing advance to 20 bits, measures the timing, and prints the result.

```
WRITEPARAM "NEW_TIME_ADV" 20
TEST TIMADV

MEASURE TIMING value
LPRINT "TIMING: " USING "2.2" value; "bits"
```

TEST TIMING

Test function : Tests the accuracy of the mobile's transmission timing at a selected timing advance

Caution : To select the timing advance to be set, the following line needs to be inserted into the program before any **TEST TIMING** command:-

WRITEPARAM "NEW_TIME_ADV" | <num> | <num var> | <num expr> |

Alternatively, the DEFAULT command will initialise the timing advance to 0 bits.

Example : The following program segment tests the timing at a timing advance of 20 bits.

```
WRITEPARAM "NEW_TIME_ADV" 20
TEST TIMING
```

Function : Timing status

Syntax : **TEST TIMING STATUS** | ON | OFF |

Note: The following parameters, **FIRSTVALUE**, **LASTVALUE**, and **INCREMENT**, have no effect if used in a **TEST TIMING** command statement (see page 3-4).

Function : Timing: first value to test in a test program

Syntax : **TEST TIMING FIRSTVALUE** | <num> | <num var> | <num expr> | BITS

Function : Timing: last value to test in a test program

Syntax : **TEST TIMING LASTVALUE** | <num> | <num var> | <num expr> | BITS

Function : Timing value increment in a test program

Syntax : **TEST TIMING INCREMENT** | <num> | <num var> | <num expr> | BITS

Function : Timing error

Syntax : **TEST TIMING ERROR** | <num> | <num var> | <num expr> | BITS

Chapter 4
INTRODUCTION TO REMOTE CONTROL

Contents

Introduction 4-1
IEEE 488.2 conventions 4-2
Command headers/compound headers 4-2
Case 4-2
Abbreviations 4-2
Program data 4-3
Response data 4-3
Terminators 4-4
Error and status reporting 4-5
Device talking formats 4-8
Message exchange protocol 4-9
IEEE488.1 Operations and states 4-9
Device Clear 4-9
Local Lockout 4-9
Reset levels 4-9
Hardware reset 4-9
Device Clear 4-10
Clear Status 4-10
Reset 4-10
Return to local 4-11
RS232 features 4-11
Handshaking 4-11
Command layout 4-11
Getting started 4-12
The remote command set 4-12
Preparing the test set for remote operation 4-12
RS232 serial port 4-12
Serial port parameters 4-12
GPIB control port 4-13
GPIB address 4-13

List of figures

Fig. 4-1 Event and Status Byte registers 4-7
Fig. 4-2 Serial port set-up menu 4-13

List of tables

Table 4-1 Error data 4-8

Introduction

The test set can be controlled remotely over either the RS232 interface or over the GPIB. The command set used is designed to comply with IEEE488.2-1987, which is a specification for GPIB.

Programs to control the test set remotely over the two interfaces will have much in common; the main difference will be the way in which characters are transmitted.

IEEE 488.2 conventions

A simple explanation of the structure of how commands and the data they take or return is presented is given here. For more complete information, refer to the latest copy of the IEEE488.2 specification.

Command headers/compound headers

Compound headers allow a complex set of commands to be built up from a smaller set of basic elements in a 'tree' structure. The elements of a compound header are separated by a colon (:).

The use of compound headers brings a number of advantages. Commands are less cryptic compared with a traditional 'flat' instrument command set.

Example:-

AFGEN1:FREQ
:LEVEL
:STATUS

Although it is possible to use the full compound header starting from the tree root every time, e.g. AFGEN1:FREQ 1KHZ;AFGEN1:STATUS ON, sequences of <COMMAND MESSAGE UNITS> and <QUERY MESSAGE UNITS> can often be shortened by taking advantage of the special rules that apply to compound headers.

Having 'descended' the tree (for example to create the <PROGRAM MESSAGE UNIT> AFGEN1:STATUS ON), any other elements at that level may be included in the <PROGRAM MESSAGE> without repeating the entire path through the tree.

Example:-

AFGEN1:FREQ 1KHZ;STATUS ON

is equivalent to the two <PROGRAM MESSAGES>

AFGEN1:FREQ 1KHZ followed by AFGEN1:STATUS ON

Note the use of the <PROGRAM MESSAGE UNIT SEPARATOR> character ";" between <PROGRAM MESSAGE UNITS>.

Here is another example:-

SCOPE:TBASE SC_500US;TRIG REPEAT

is equivalent to the two <PROGRAM MESSAGES>

SCOPE:TBASE SC_500US and SCOPE:TRIG REPEAT

To return to the top of the tree so that another 'branch' may be descended, a colon is used.

Example:-

MODGEN1:FREQ 10KHZ;LEVEL 100MV;:MODGEN2:FREQ 3KHZ

Case

The software is not case-sensitive. Upper and lower case characters are completely interchangeable. There is no conflict between milli (m) and mega (M) as both cannot be applied to the same data.

Abbreviations

In general, header elements can be abbreviated to the shortest unique string at that level and part of the command tree.

Example:-

AFGEN1:F is equivalent to AFGEN1:FREQ

Program data

The following program data functional elements are accepted by the test set:-

<CPD> (also known as <CHARACTER PROGRAM DATA>)
<NRf> (also known as <DECIMAL NUMERIC PROGRAM DATA>)
<STRING PROGRAM DATA>

All these functional elements are defined in IEEE 488.2-1987.

<CPD>

Character program data is used to set a parameter to one of a number of states that are best described by short alphanumeric strings.

Example:-

ON

OFF and ON are the possible <CPD> elements to set the status of the RF generator. Note that the short form (i.e. OF or ON) may be used when setting the parameter.

<NRf>

Flexible numeric representation (also known as <DECIMAL NUMERIC PROGRAM DATA>) covers integer and floating point representations.

Examples:-

–466	Integer value
4.91	Explicitly placed decimal point
59.5E+2	Mantissa and Exponent representation

The format is known as 'flexible' because any of the three representations may be used for any type of numeric parameter.

Examples:-

Where a parameter requires an integer value in the range 1 to 100, and the user needs to set its value to 42, the following values will be accepted by the test set:-

42	Integer
42.0	Floating point
4.2E1, 4200E-2	Floating point - Mantissa/exponent
41.5	Rounded up to 42
42.4	Rounded down to 42

<STRING PROGRAM DATA>

String program data consists of a number of ASCII characters enclosed in quotes. Either a pair of single ('ASCII 39') or double ("ASCII 34") quotes may be used. If the quote character chosen to mark the beginning and end of the string also appears within it, it must be doubled.

Example:-

'This string contains the word "Hello"'

will be interpreted as the string

This string contains the word 'Hello'

Response data

The following response data functional elements are generated by the test set:-

<CRD> (also known as <CHARACTER RESPONSE DATA>)
<NR1>
<NR2>
<NR3>
<STRING RESPONSE DATA>

<CRD>

This type of response is returned when reading the value of a parameter which can take a number of discrete states. States are represented by short alphanumeric strings.

Example:-

ON

OFF and ON are the possible <CRD> responses if the parameter that determines the status of the RF frequency generator is queried.

Note that the short form (i.e. OF and ON) may be used when setting the parameter. When the parameter is queried, the long form is always returned.

<NR1>

This type of numeric response is used when returning the value of integer parameters, such as averaging number or number of measurement points.

Examples:-

15
+3
–57

<NR2>

This type of numeric response includes an explicitly placed decimal point, but no exponent.

Examples:

17.91
–18.27
+18.83

<NR3>

This type of numeric response includes an explicitly placed decimal point and an exponent.

Examples:

1.756E+2
182.8E–3

<STRING RESPONSE DATA>

This takes a similar form to <STRING PROGRAM DATA> except that the delimiting character is always a double quote ("ASCII 34").

Terminators

A **<PROGRAM MESSAGE TERMINATOR>** (as defined in IEEE 488.2-1987) can be a newline character (ASCII 10), a newline character with the ^END message asserted at the same time, or an ^END message asserted with the final character of the <PROGRAM MESSAGE>. The terminator may be preceded by any number of 'white space' characters - i.e., any single ASCII-encoded byte in the range 0 to 9 and 11 to 32 decimal.

A **<RESPONSE MESSAGE TERMINATOR>** (as defined in IEEE 488.2-1987) is a newline character with the ^END message asserted at the same time.

Many GPIB controllers terminate program messages with a newline character and, by default, accept newline as the response message terminator. When transferring binary data, which may contain embedded newline characters, it is necessary to ensure that the controller uses only ^END messages. Usually this means that the controller's GPIB must be set up to generate and detect ^END. Refer to the documentation supplied with the controller.

Error and status reporting

Status reporting is implemented in accordance with IEEE 488.2

There are two status reporting bytes; the **Standard Event Status** register and the **Status Byte** register. Each has an associated mask. At power on, both masks are set to 0, i.e., all bits are disabled.

Standard Event Status register

The eight bits in this register indicate the following conditions:-

Bit 0 Operation Complete - Generated in response to a *OPC command. Indicates that the device has completed all selected pending operations.

Bit 1 Controller in charge - This function is not relevant to this test set and is not supported.

Note that this function is currently not used.

Bit 2 Query Error (QYE) - This indicates that either:-

(a) An attempt is being made to read data from the output queue when no output is present or pending.

(b) Data in the output queue has been lost.

Bit 3 Device Specific Error (DDE) - A device specific error, such as out of range data has been requested.

Bit 4 Execution Error (EXE) - Execution errors are detected by the execution control block. This event bit indicates that either:-

(a) Data was sent that was incompatible with the device.

(b) A valid program message could not be executed due to some device condition.

Bit 5 Command Error (CME) - Detected by the parser. Illegal syntax, or unknown headers are indicated by this event.

Bit 6 User Request (URQ) - Indicates that one of a set of local controls has been activated.

Bit 7 Power On (PON) - This event bit is set to indicate that a off-to-on transition has occurred in the device's power supply; i.e., this bit gets set once the device has cleared and the GPIB has been initialised and is ready to receive data.

This register is *destructively* read by *ESR?.

Bits 2, 3, 4 and 5, the error bits, are cleared on a read and the particular errors are read by the error commands COMerror?, DEVerror?, Execerror?, and Qerror?. See Table 4-1 on page 4-8, and Chapter 5.

The **Standard Event Status Enable** register is a mask for this register which is read using *ESE? and can be set using *ESE <NR1>. The power-on value for this mask is 0. It is used to create the **Event Summary Bit**. The Standard Event Status Enable register is not cleared by *CLS, *RST, Device Clear, or a return to local operation. The only way to clear the mask is to set it directly or by performing a hardware reset (page 4-9).

Status Byte register

This register contains the following bits:-

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.
Bit 3 Not used.
Bit 4 Message Available (MAV).
Bit 5 Event Summary Bit.
Bit 6 MSS (Master Summary Status) *or* RQS when read with serial poll.
Bit 7 Not used.

The register is *non-destructively* read by *STB?, or a serial poll, although a serial poll clears the RQS bit.

The Master Summary Status (MSS) is the combination of the other bits in the Status Byte, and the **Service Request Enable** register. This can be read with *SRE? or set with *SRE <NR1>. The Service Request Enable register is not cleared by a *CLS, *RST, Device Clear, or a return to local operation. The only way to clear the mask is to set it directly or by performing a hardware reset (page 4-9).

The RQS (ReQuest for Service) is the IEEE 488.1 interrupt line. It is raised whenever a bit in the Status Byte is set to 1, and its corresponding enable bit is set. RQS is cleared by a serial poll, the return value of which represents the Status Byte value, with RQS in place of MSS. Subsequent serial polls will return the Status Byte value less 64, as RQS will no longer be set to 1. MSS is not affected by a serial poll.

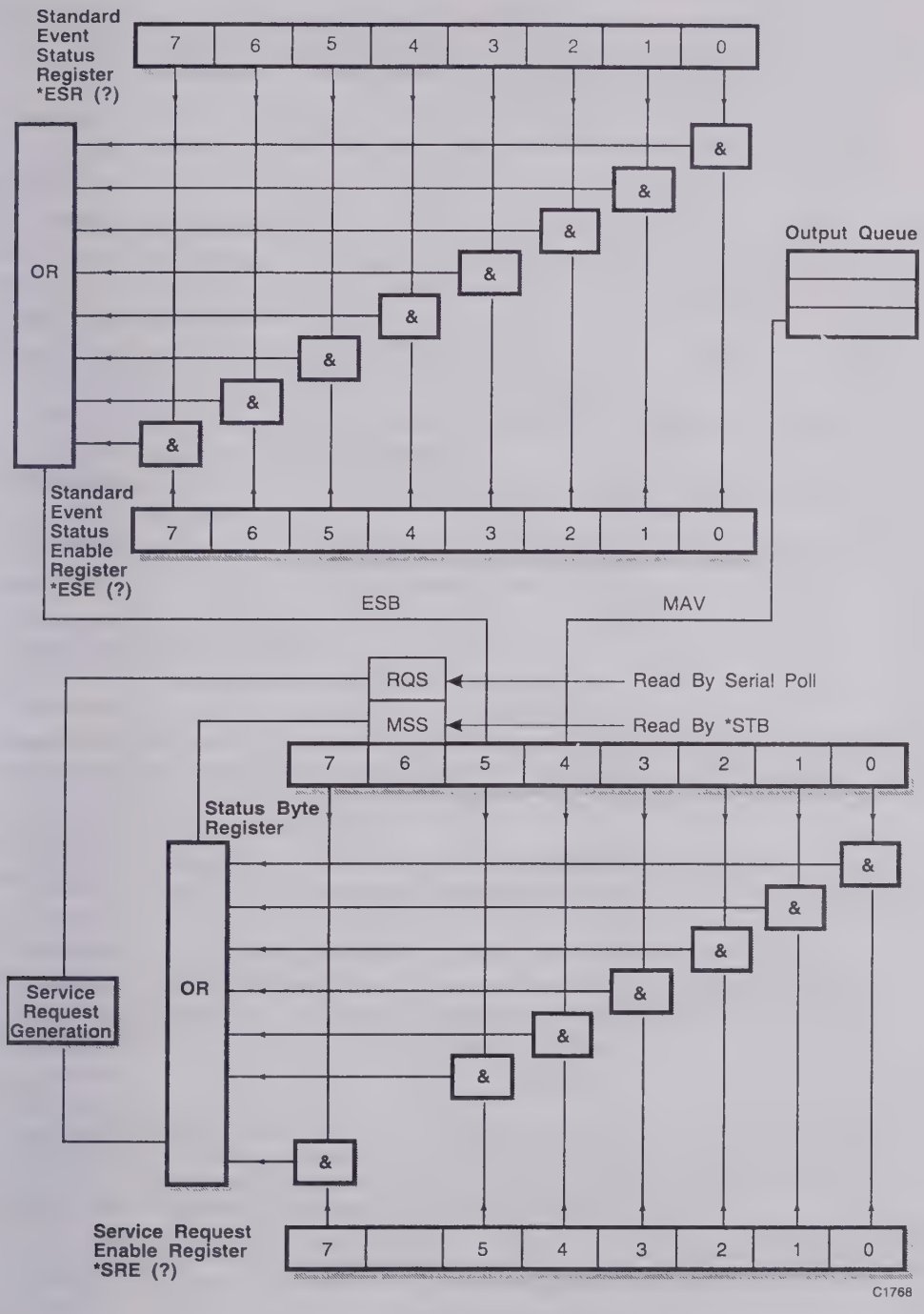


Fig. 4-1 Event and Status Byte registers

Table 4-1 Error data

<QUERY MESSAGE UNIT>	Error data	Indication
COMmerror?	0	'No Error'
	1	'Illegal Star Command'
	2	'Sub Command Not Allowed'
	3	'No Header Match Found'
	4	'Header Text Is Not Unique'
	5	'Action Not Allowed With Header'
	6	'Query Not Allowed With Header'
	7	'Parser Decode Error'
	8	'Header Requires Sub-Header'
DEVerror?	0	'No Error'
	1	'Instrument Range or Setting Error'
	2	'Request For Measurement Pending'
	3	'Calibration already running'
	4	'Systems Test already running'
	5	'Digital measurements not valid'
	6	'Measurement invalid in this mode'
	7	'Frequency Standard Error'
EXecerror?	0	'No Error'
	1	'Data Type Not Allowed'
	2	'Excess Data'
	3	'Insufficient Data'
	4	'Some Data Required'
	5	'No Text Match Found'
	6	'Text Not Unique'
	7	'Suffix Not Found'
	8	'Suffix Not Allowed'
	9	'Query Error'
Qerror?	0	'No Error'
	1	'Interrupted'
	2	'Unterminated'
	3	'Deadlocked'

Device talking formats

Query response formats

The response to a standard query (where allowed) is as follows:-

Main <*program mnemonic*>[:Subsidiary <*program mnemonic*>]<*data*>[<*units*>]

The Main <*program mnemonic*> and :Subsidiary <*program mnemonic*> are returned in the format defined by the command:-

RESPONSE:HEADER

The data is in the appropriate format, and the units are for maximum resolution, i.e., the minimum units allowable for settings. This means that, for example, the RF generator frequency is returned in Hz, not MHz. Full details of the units used can be found in Chapters 5 and 6.

A query returns the current value of the setting, places the result in the output queue and raises the message available (MAV) bit in the Status Byte. See under *Error and status reporting* on page 4-5.

Message exchange protocol

IEEE488.2 defines a protocol for the exchange of messages between devices. There are three errors states that the test set can enter should this protocol be broken. The error states and the reasons that they occur are:-

1. **UNTERMINATED** occurs when the controller attempts to read a response without having sent a complete query.
2. **INTERRUPTED** occurs when the controller starts to send a new message before having read the response to a preceding terminated query.
3. **DEADLOCK** occurs if the input and output buffers both become full, and the controller is still trying to send a byte.

The test set has input and output buffers of 8 kbytes length. The output buffer is effectively full if there is insufficient room for it to contain the next formatted message.

IEEE488.1 Operations and states

Device Clear

Device Clear is an operation defined over the GPIB bus. Upon receiving a Device Clear, the test set is sent into the remote state, clears both its input and output buffers, and resets the remote software to a known state. It does not alter the state of any flags within the status reporting other than MAV.

The main use of Device Clear is to reset the communication when there has been any communication problem. It is good practice to send a Device Clear at the beginning of a remote program.

Note: the IEEE 488.2 Device Clear function only affects the GPIB functions. The input and output buffers are cleared and the test set put into a state to accept new messages. It does not put the test set functions into a defined state; this is performed by the *RST common command (page 4-10).

Local Lockout

Over Remote, the controller can set the test set into **Local Lockout** state. When Local Lockout is set, the front panel is disabled and the [HELP SET-UP] (LOCAL) key is made ineffective. Local lockout is often used when the test set is part of an automatic test system and left unattended. In this state, the test set cannot be affected by operations from the front panel.

Sending the test set local over the remote does not release this state. The keyboard can only be re-enabled by releasing Local Lockout over the remote interface or by switching the power supply off and on.

Reset levels

Hardware reset

The highest level of reset is the full hardware reset. This is actioned by issuing the command:-

HWRESET

This is the same as switching the test set OFF and ON again.

Actions

All test set hardware is reset

The GPIB hardware interface is reset

The message buffers are cleared

The Standard Event Status and Status Byte registers are reset to 0

The Standard Event Status Enable and Service Request Enable registers are reset to 0

The measurement cycle is reset, and pending measurements are cleared

The test set returns to local

All test modes are reset to the power-up default values

The power-up test mode is selected

The PON flag is set to 1

Device Clear

This is actioned by issuing an IEE488.1 interface clear message.

Actions

The Input and Output buffers are cleared

The MAV bit in the Status Byte is set to 0

Pending operations are cleared

Pending OPC requests are cleared

Pending measurement requests are cleared

The Standard Event Status register is not affected

The Status Byte is not affected (except for MAV)

The Standard Event Status Enable and Service Request Enable registers are not affected

Clear Status

This is actioned by issuing the command:-

*CLS

Actions

The Status Byte is reset to 0, unless there is a message available, in which case the MAV is set to 1

The Standard Event Status register is set to 0

The Standard Event Status Enable and Service Request Enable registers are unaffected

Reset

This is actioned by issuing the command:-

*RST

Actions

Pending measurement requests are cleared

Pending OPC requests are cleared

All test modes are reset to the power-up default values

The power-up test mode is selected

The Standard Event Status register is not affected

The Status Byte is not affected

The Standard Event Status Enable and Service Request Enable registers are not affected

Return to local

Actioned by either sending the IEE488.1 interface message Go to local, or by pressing the [HELP SET-UP] key when local lockout is not activated.

Actions

The test set returns to local operation

The Standard Event Status register is not affected

The Status Byte is not affected

The Standard Event Status Enable and Service Request Enable registers are not affected

The output queue is not affected

The input queue is not affected

Note

The information given above is correct to the IEEE 488.2 specification, to the best of Marconi Instruments' knowledge. The test set is known to conform to the details given above. Other areas of the manual may appear to imply differing operation. If this is the case, then the operation described in this section should be regarded as correct.

RS232 features

Handshaking

Handshaking of communication over the GPIB is automatic, but over the RS232 the test set implements it by using the handshake characters XON (start transmitting) and XOFF (stop transmitting).

Command layout

In the lists of commands in Chapters 5 and 6, each command is set out as follows:-

1. **Path from the subsystem root**

   Example:-

   ```
   :RFGen
      :FREQ
   ```

2. **Description**

   Describes the purpose of the command.

3. **Parameters**

   Where applicable, each parameter is listed, stating its <PROGRAM DATA> functional element (as defined in IEEE 488.2-1987) and possible values.

   Example:-

   <CPD> OFF, ON

4. **Allowed suffices**

   A list of the suffices or units allowed for numeric values is provided.

   Example:-

   ```
   MHz, kHz
   ```

   This means that a frequency can be entered with either MHz or kHz units.

5. **Example**

   An example of the use of the command is provided.

6. **Response**

   Query responses shows the response in terms of its IEEE 488.2 functional elements, and the semantics of the response.

   Example:-

   <NR2> Frequency in Hz

7. **Example response**

   This field gives an example of a typical response from a query. Usually this corresponds to the example field.

Getting started

This section provides an introduction to the remote programming of the test set, including a worked example.

The remote command set

The first point to notice when controlling the test set over one of the remote interfaces is that there is not a straightforward mapping between manual front panel operations and their remote equivalents.

Common commands

The IEEE 488.2 common commands all start with a "*" character. Those that are implemented in the test set are listed in Chapter 5.

The most important command is *RST, which places the test set in a defined state. It is good practice to send *RST at the start of any remote program.

Preparing the test set for remote operation

RS232 serial port

The connections required between the RS232 serial port and the controlling device are described in Chapter 2, *Installation*, of the Operating Manual, in the section *Serial port*.

Serial port parameters

The remote control facilities on the test set allow a variety of input and output combinations of GPIB and RS232. These are selected from the SET-UP I/O: REMOTE OPTIONS menu, which is reached using the following key sequence:-

[HELP SET-UP], *[SET-UP]*, *[INPUT OUTPUT]*, *[remote options]*

Repeatedly pressing the *[remote input]* key will scroll through the options of:-

RS232
GPIB
RS232 or GPIB

Repeatedly pressing the *[remote output]* key will alternate between the options of:-

RS232
GPIB

If RS232 is to be used for either function, the serial port must be configured to match the serial port configuration of the controlling device. This is done from the SET-UP I/O: SERIAL PORT (RS232 / EIA-232) menu (see Fig. 4-2), reached using the following key sequence:-

[HELP SET-UP], *[SET-UP]*, *[INPUT OUTPUT]*, *[serial set-up]*

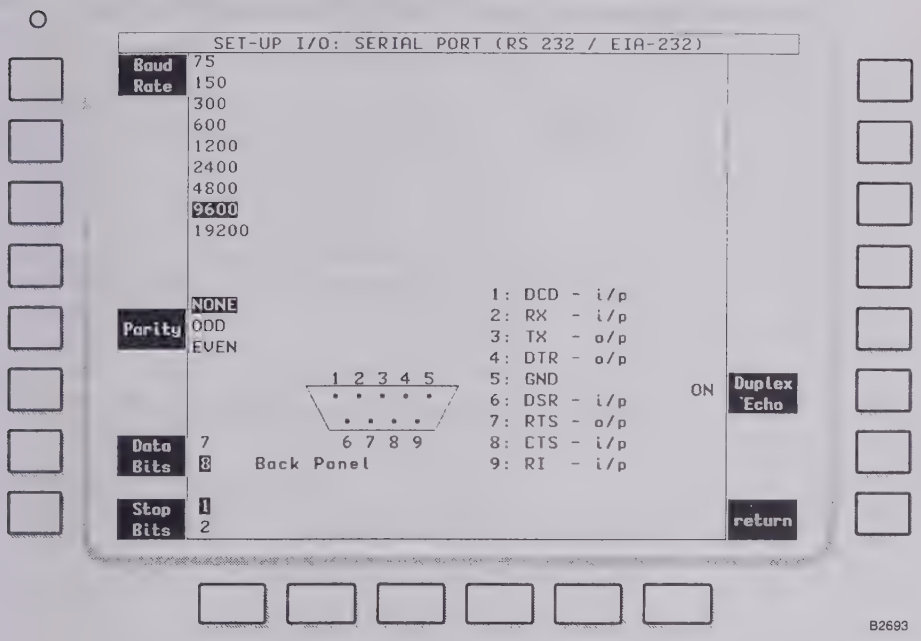


Fig. 4-2 Serial port set-up menu

Under most operating conditions, the default settings are the performance optima. These should be used unless the control device requires a different setting.

Repeatedly pressing the appropriate soft key will scroll through the options. The default settings are:-

Baud Rate: 9600
Parity: NONE
Data Bits: 8
Stop Bits: 1

The *[duplex echo]* key on this menu controls whether or not any incoming data from the serial port is echoed back to the transmitting terminal as confirmation.

GPIB control port

The connections required between the GPIB port and the controlling device are described in Chapter 2, *Installation*, of the Operating Manual, in the section *GPIB port*.

GPIB address

The test set must be given an address code before it can be used by remote control over the GPIB. This address is entered on the SET-UP I/O: GPIB (IEEE 488) menu, reached by the key sequence [HELP SET-UP], *[SET-UP]*, *[INPUT OUTPUT]*, *[GPIB set-up]*. Press *[address]* and then enter the required address number using the data entry keys. The number must be unique on the system to the test set and within the range 1 to 30.

Example - simple receiver final test

In this example, remote commands are stated without making any assumptions about the controller and programming language to be used. These commands, of course, will need to be incorporated into the program language statements of the target controller. Here are some examples of how this would be done in practice, using the reset command *RST. The test set address is assumed to be 6.

```
PRINT @6:"*RST"
```
Controller using TBASICR programming language (TransEra Corporation).

```
OUTPUT 706;"*RST"
```
Controller using HTBASICTM programming language (TransEra Corporation).

It may sometimes be necessary to send a Device Clear command, if the GPIB system fails to respond to *RST or appears to lock up. Examples are as follows:-

```
WRITE GPIB CMD_SDC(6)
```
Controller using TBASICR.

```
CLEAR 706
```
Controller using HTBASICTM.

Step 1. Preset the test set to a known state

```
DEVICE CLEAR
*RST
```
Preset the test set.

Step 2. Select the test set mode for the test

```
TEST RX
```
Select Receiver test mode.
Remember that IEEE 488.2 requires a single space character between the command header and its parameter(s).

Step 3. Set RF output port, frequency and level

```
RFPORT N_OUT_SIMPLEX
```
Select the N-type output port.

```
RFGEN:FREQ 470.0MHZ
```
Set the test set signal generator frequency to 470 MHz.

```
RFGEN:LEV -110DBM
```
Set the signal generator level to -110 dBm.

Step 4. Set mod type, mod gen level

```
MODTYPE FM
```
Generate frequency modulation.

```
MODGEN2:FMDEVN 6KHZ;STATUS ON
```
Set mod gen 2 deviation to 6 kHz and switch on.

Step 5. Set distortion measurement type

```
RXDISTN SINAD
```
Select the measurement of receiver distortion to be SINAD. A requested measurement of SINAD will cause an error if the correct measurement type is not selected.

Step 6. Measure audio level

```
MEASU:AFLEVEL?
```
Take a measurement of audio level at the front panel AF input.

Step 7. Measure audio frequency

```
MEASU:AFFREQ?
```
Measure the frequency of the audio signal.

Step 8. Measure audio SINAD

```
MEASU:RXSINAD?
```
Request a measurement of receiver SINAD (at the AF input).

Chapter 5
INSTRUMENT COMMANDS

Contents

Introduction 5-1
Command abbreviations 5-1
Common commands 5-1
List of commands. 5-1
Instrument measurement commands 5-6
List of commands 5-6
Measurement cycle 5-6
Summary of measurement response formats 5-15
Instrument digital measurement commands 5-17
List of commands 5-17
Measurement cycle 5-17
Summary of digital measurement response formats 5-24
Memory and memory card commands 5-25
List of commands 5-26
Instrument-specific commands 5-30
List of commands 5-30
Tones operation commands 5-187
List of commands. 5-187

Introduction

This chapter describes the instrument commands, that is, those commands that can be used to directly control instrument settings and measurements. Chapter 6 describes Systems commands, that is, those commands that are specifically intended for use in System test programs.

Command abbreviations

The command syntax provided in this chapter and in Chapter 6 indicates the minimum number of characters that must be entered for a command to be valid by showing them in uppercase. For example, the command :BArchart can be entered as just :BA. Conversely, the complete command, :BARCHART, is equally valid, and more readily understandable. Another consideration is that the addition of new commands in future development could invalidate the currently acceptable minimum number of characters. It is therefore recommended that, where possible, the complete command is used.

Common commands

List of commands

The following common * (star) commands are implemented:-

***Cls**	***OPC**	***SRe**
***ESE**	***OPC?**	***SRe?**
***ESE?**	***OPT?**	***STb?**
***ESR?**	***Rst**	***TSt?**
***Idn?**		

*Cls

Description : Clear Status command. The Status Byte is set to 0, unless there is a message available, in which case the MAV is set to 1. The Standard Event Status register is set to 0. The Standard Event Status Enable and Service Request Enable registers are unaffected. See also the *Reset levels* section in Chapter 4.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `*Cls`
Clears Status registers as described above

*ESE

Description : Event Status Enable command. Sets the Standard Event Status Enable register. The range is 0 to 255.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `*ESE 255`
Sets the Standard Event Status Enable register to 255

*ESE?

Description : Event Status Enable query. Returns the value of the Standard Event Status Enable register as NR1. The range of the returned data is 0 to 255.

Response : <NR1>

Example : `*ESE?`

Example response : `255`

*ESR?

Description : Event Status Register query. Returns the value of the Standard Event Status register. The range of the returned data is 0 to 255. This is a destructive read which clears the register and associated summary bits.

Response : <NR1>

Example : `*ESR?`

Example response : `8`

*Idn?

Description : Identification query. Returns an arbitrary ASCII response comprising four data fields in the format:-

<Manufacturer>,
<type number>,
<serial number>,
<firmware version number and date>:<sub-firmware version A1>:
<sub-firmware version A5>:<sub-firmware version A7>:
<sub-firmware version AE1>:<sub-firmware version A4><EOM>
Note: If A4 is not fitted, response is 0.00.

Response : <Arbitrary ASCII response data>, <Arbitrary ASCII response data>, <Arbitrary ASCII response data>, <Arbitrary ASCII response data>.

Example : `*Idn?`

Example response : `MI,2966A,132637-001,`
`6.01 23/03/95:5.20:6.00:5.00:5.02:1.00<EOM>`

*OPC

Description : Operation Complete command. Sets the Operation Complete bit in the Standard Event Status register when execution of all preceding operations are complete.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `*OPC`
Sets Operation Complete bit when all operations have been completed

*OPC?

Description : Operation Complete query. Returns a 1 if all the preceding operations have been completed.

Response : <NR1>

Example : `*OPC?`

Example response : `1`

*OPT?

Description : Returns the state of enabled options in 30 data fields.

The first 10 are for normal instrument options
The second 10 are for analogue cellular options
The third 10 are for digital cellular options

Response : <NR1> Data field 1, state of MAINCODE OPTION
Data field 2, state of SSB OPTION
Data fields 3 to10, reserved for future use; returns 0
Data field 11, state of PMR OPTION
Data field 12, state of MULTI-AMPS OPTION
Data field 13, state of MULTI-TACS OPTION
Data field 14, state of NMT450 900 OPTION
Data field 15, state of MPT1327 OPTION
Data fields 16 to 20, reserved for future use; returns 0
Data field 21, state of GSM OPTION
Data fields 22 to 30, reserved for future use; returns 0

Example : `*OPT?`

Example response : `1,0,0,0,0,0,0,0,0,0,1,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,`

*Rst

Description : Reset command. Pending measurement requests are cleared. Pending OPC requests are cleared. All test modes are reset to the power-up default values. The power-up test mode is selected. The Standard Event Status register is not affected. The Status Byte is not affected. The Standard Event Status Enable and Service Request Enable registers are not affected. See also the *Reset levels* section in Chapter 4.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `*Rst`
Sets instrument to known state

*SRe

Description : Service Request Enable command. Sets the Service Request Enable register. Range 0 to 255.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `*SRe 32`
Sets Service Request Enable register

*SRe?

Description : Service Request Enable query. Returns the value of the Service Request Enable register as NR1.

Response : <NR1>

Example : `*SRe?`

Example response : `32`

*STb?

Description : Status Byte query. Returns the value of the Status Byte.

Response : <NR1>

Example : `*STb?`

Example response : `32`

*TSt?

Description : Self Test query. Returns a 0 if the instrument passed all self tests, or 1 if there was a failure.

Response : <NR1>

Example : `*TSt?`

Example response : `0`

Instrument measurement commands

List of commands

:MEASUre
- :AFFreq?
- :AFLevel?
- :AMDepth?
- :AMPs?
- :FMDevn?
- :FMLevels?
- :FWdpwr?
- :MKR1?
- :MKR2?
- :MKRDiff?
- :MOdfreq?
- :Ohms?
- :PMDevn?
- :PMLevels?
- :REvpwr?
- :RXDistn?
- :RXSInad?
- :RXSN?
- :Ssbsuppression?
- :TXDistn?
- :TXFreq?
- :TXLevel?
- :TXOffset?
- :TXReceived?
- :TXSn?
- :VOlts?
- :VSwr?

Measurement cycle

The measurement speed is affected by the measurement cycle mode, which determines the way in which measurements are returned over the remote control. Briefly, if the mode is ON, measurements are returned more quickly, whereas when it is OFF there is more certainty that measurements are current. For more information, refer to the description of the :MEASCYCLE command on page 5-112.

:MEASUre

:AFFreq?

Description : Returns the value of audio frequency.

Response : <NR2> Frequency in Hz

Example : `:MEASUre:AFFreq?`

Example response : `999.1`
Measured audio frequency is 999.1 Hz

:MEASUre

:AFLevel?

Description : Returns the value of audio level.

Response : <NR2> Level in mV

Example : `:MEASUre:AFLevel?`

Example response : `101.1`
Measured amplitude of audio signal is 101.1 mV

:MEASUre

:AMDepth?

Description : Returns the value of amplitude modulation depth.

Response : <NR2> Depth in %

Example : `:MEASUre:AMDepth?`

Example response : `25.1`
The measured amplitude modulation depth is 25.1%

:MEASUre

:AMPs?

Description : Returns the value of current through the terminals of the multimeter.

Response : <NR2> Units in Amperes

Example : `:MEASUre:AMPs?`

Example response : `0.10100`
Measured current is 101 mA

:MEASUre

:FMDevn?

Description : Returns the value of frequency modulation deviation.

Response : <NR2> Frequency in Hz

Example : `:MEASUre:FMDevn?`

Example response : `1523.4`
The measured frequency modulation deviation is 1523.4 Hz

:MEASUre

:FMLevels?

Description : Returns the values of positive frequency deviation and negative frequency deviation.

Response : <NR2>,<NR2> Pos frequency in Hz, Neg frequency in Hz

Example : `:MEASUre:FMLevels?`

Example response : `1523.4,1534.2`
The positive deviation is 1523.4 Hz
The negative deviation is 1534.2 Hz

:MEASUre

:FWdpwr?

Description : Returns the value of forward power taken by the directional power head accessory.

Response : <NR2> Units in mW

Example : `:MEASUre:FWdpwr?`

Example response : `45`
The measured forward power is 45 mW

:MEASUre

:MKR1?

Description : The response to the Measurement Marker 1 command is dependent upon the current display type selection. The table below shows the returned format and the units or representation, for each of the possible display types.

Response :

Spectrum analyzer	<NR1>		Level in dBm
Audio oscilloscope	<NR1>	<NR1>	Max, Min level
Mod oscilloscope	<NR1>	<NR1>	Max, Min level
Audio FFT analyzer	<NR2>		Level in µV
Mod FFT analyzer	<NR2>		FM in Hz AM in % PM in rad
CW analyzer or sideband analyzer	<NR2>		Relative level in dB
Digital power analyzer	<NR2>		Relative level in dB
Digital phase analyzer	<NR2>		Phase in degrees

Example : `:MEASure:MKR1?`

Example response : `15.000`
If the modulation FFT analyzer is in use and FM demod is being used, Marker 1 is indicating a level of 15 kHz

:MEASUre

:MKR2?

Description : The response to the Measurement Marker 2 command is dependent upon the current display type selection. The table below shows the returned format and the units or representation, for each of the possible display types.

Response :

Spectrum analyzer	<NR2>		Level in dBm
Audio oscilloscope	<NR1>	<NR1>	Max, Min level
Mod oscilloscope	<NR1>	<NR1>	Max, Min level
Audio FFT analyzer	<NR2>		Level in µV
Mod FFT analyzer	<NR2>		FM in Hz AM in % PM in rad
CW analyzer or sideband analyzer	<NR2>		Relative level in dB
Digital power analyzer	<NR2>		Relative level in dB
Digital phase analyzer	<NR2>		Phase in degrees

Example : `:MEASure:MKR2?`

Example response : `20.1`

If the spectrum analyzer is in use, Marker 2 is indicating a level of 20.1 dBm.

:MEASUre

:MKRDiff?

Description : The response to the Measurement Marker Difference command is dependent upon the current display type selection. The table below shows the returned format and the units or representation, for each of the possible display types.

Response :

Spectrum analyzer	<NR2>	Difference in dB
Audio oscilloscope	<NR2>	Difference in mV
Mod oscilloscope	<NR2>	Difference:- FM in Hz AM in % PM in rad
Audio FFT analyzer	<NR2><NR2>	Difference in µV, dB
Mod FFT analyzer	<NR2><NR2>	Difference:- FM in Hz, dB AM in %, dB PM in rad, dB
CW analyzer or sideband analyzer	<NR2>	Difference in dB
Digital power analyzer	<NR2>	Difference in dB
Digital phase analyzer	<NR2>	Difference in degrees

Example : `:MEASure:MKRDiff?`

Example response : `2000`

If the audio oscilloscope is in use, the difference between Marker 1 level and Marker 2 level is 2 V.

:MEASUre

:MOdfreq?

Description : Returns the value of modulation frequency.

Response : <NR2> Frequency in Hz

Example : `:MEASUre:MOdfreq?`

Example response : `999.0`

The measured modulation frequency is 999 Hz

:MEASUre

:Ohms?

Description : Returns the value of resistance across the multimeter terminals.

Response : <NR2> Units of Ohms

Example : `:MEASUre:Ohms?`

Example response : `2500.00`

The measured resistance is 2.5 kΩ

:MEASUre

:PMDevn?

Description : Returns the value of phase modulation.

Response : <NR2> Deviation in radians

Example : `:MEASUre:PMDevn?`

Example response : `1.30`
Phase deviation is 1.3 radians

:MEASUre

:PMLevels?

Description : Returns the values of positive phase modulation and negative phase modulation.

Response : <NR2>,<NR2> Pos deviation in rad, Neg deviation in rad

Example : `:MEASUre:PMLevels?`

Example response : `2.10;2.50`
The positive phase deviation is +2.1 rad.
The negative phase deviation is 2.5 rad.

:MEASUre

:REvpwr?

Description : Returns the value of reverse power taken by the directional power head accessory.

Response : <NR2> Units in mW

Example : `:MEASUre:REvpwr?`

Example response : `3`
The measured reverse power is 3 mW

:MEASUre

:RXDistn?

Description : Returns the value of Rx distortion.

Response : <NR2> distortion in %

Example : `:MEASUre:RXDistn?`

Example response : `1.5`
The measured distortion is 1.5%

:MEASUre

:RXSInad?

Description : Returns the value of Rx SINAD.

Response : <NR2> Sinad in dB

Example : `:MEASUre:RXSInad?`

Example response : `26.0`
Measured value of SINAD is 26 dB

:MEASUre

:RXSN?

Description : Returns the value of Rx signal-to-noise ratio.

Response : <NR2> Signal to noise in dB

Example : `:MEASUre:RXSN`

Example response : `28.0`
Measured receiver signal-to-noise ratio is 28 dB

:MEASUre

:Ssbsuppression?

Description : Returns the values for SSB suppression measurements taken by the directional power head accessory.

Response : <NR2>,<NR2>,<NR2>,<NR2> Units of mW, dBr, dBr, dBr

Example : `:MEASUre:Ssbsuppression?`

Example response : `10.1, 0.0, -60.0, -45.0`
The absolute power level is 10.1 mW
The lower sideband relative power is 0.0 dBr
The carrier wave relative power is –60.0 dBr
The upper sideband relative power is –45.0 dBr
where the relative levels are relative to the absolute power level.

:MEASUre

:TXDistn?

Description : Returns the value of Tx distortion.

Response : <NR2> Distortion in %

Example : `:MEASUre:TXDistn?`

Example response : `2.1`
The measured distortion is 2.1%

:MEASUre

:TXFreq?

Description : Returns the value of Tx frequency.

Response : <NR2> Frequency in MHz

Example : `:MEASUre:TXFreq?`

Example response : `101.537123`
The measured frequency of the transmitter is 101.537123 MHz

:MEASUre

:TXLevel?

Description : Returns the value of Tx level.

Response : <NR2> Level in mW for broad-band; dBm for in-band.

Example : `:MEASUre:TXLevel?`

Example response : `31.200`
Returns the Tx Level in mW or dBm.

:MEASUre

:TXOffset?

Description : Returns the value of Tx frequency offset.

Response : <NR2> Frequency in Hz

Example : `:MEASUre:TXOffset?`

Example response : `30.000`
The measured offset frequency is 30.000 Hz

:MEASUre

:TXReceived?

Description : Returns the values of Tx frequency and Tx offset.

Response : <NR2>,<NR2> TX freq in MHz, TX offset in Hz

Example : `:MEASUre:TXReceived?`

Example response : `890.123456,30`
The current measured RF value is 890.123456 MHz and the current measured offset from the set Tx frequency is 30 Hz

:MEASUre

:TXSn?

Description : Returns the value of Tx signal-to-noise ratio.

Response : <NR2> Signal-to-noise ratio in dB

Example : `:MEASUre:TXSn?`

Example response : `21.0`
The measured signal-to-noise ratio is 21.0 dB

:MEASUre

:VOlts?

Description : Returns the value of voltage present at the multimeter terminals.

Response : <NR2> Units in Volts

Example : `:MEASUre:VOlts?`

Example response : `3.000`
Measured voltage is 3 V

:MEASUre

:VSwr?

Description : Returns the value of voltage standing wave ratio, taken by the directional power head accessory.

Response : <NR2>

Example : `:MEASUre:VSwr?`

Example response : `1.1`
The measured voltage standing wave ratio is 1.1.

Summary of measurement response formats

The measurements are returned as data, with no unit field, in the following formats:-

Measurement	Returned Format	Units/Representation
:AFFreq?	<NR2>	Hz
:AFLevel?	<NR2>	Dependent upon present unit of measurement: mV / dBV / dBm / dBr
:AMDepth?	<NR2>	%
:AMPs?	<NR2>	A
:FMDevn?	<NR2>	Hz
:FMLevels?	<NR2>,<NR2>	Pos devn Hz, neg devn Hz
:FWdpwr?	<NR2>	mW
:MOdfreq?	<NR2>	Hz
:Ohms?	<NR2>	Ω
:PMDevn?	<NR2>	rad
:PMLevels?	<NR2>,<NR2>	Pos devn rad, neg devn rad
:REvpwr?	<NR2>	mW
:RXDistn?	<NR2>	%
:RXSInad?	<NR2>	dB
:RXSN?	<NR2>	dB
:Ssbsuppression?	<NR2>,<NR2>,<NR2>,<NR2>	mW, LSBdBr, CWdBr, USBdBr
:TXDistn?	<NR2>	%
:TXFreq?	<NR2>	MHz
:TXLevel?	2<NR2>	Brdband mW, Nrwbnd dBm
:TXOffset?	<NR2>	Hz
:TXReceived?	<NR2>,<NR2>	Freq MHz, offset Hz
:TXSn?	<NR2>	dB
:Volts?	<NR2>	V
:VSwr?	<NR2>	No units (ratio to 1)

Markers are dependent upon the current display type selected (see below and page 5-15).

Spectrum analyzer

:MKR1?	<NR2>	dBm
:MKR2?	<NR2>	dBm
:MKRDiff?	<NR2>	dB

Oscilloscope

Absolute marker levels do not have any relevance as they are not relative to any point (since the trace is movable). They are, however, returned, as they can be used, for example, to find the peak over remote by moving the marker along in small increments and returning the marker level. Two values are returned, since the trace may have noise on it, and a single ordinate contains more than a single trace point.

MKRDiff does have a value that is relevant, and it is returned in the appropriate units.

Audio oscilloscope

:MKR1?	<NR1>,<NR1>	Max, Min level
:MKR2?	<NR1>,<NR1>	Max, Min level
:MKRDiff?	<NR2>	mV

Mod oscilloscope

:MKR1?	<NR1>,<NR1>	Max, Min level
:MKR2?	<NR1>,<NR1>	Max, Min level
:MKRDiff?	<NR2>	FM in Hz, AM in %, PM in rad

Audio FFT analyzer

:MKR1?	<NR2>	µV
:MKR2?	<NR2>	µV
:MKRDiff?	<NR2>, <NR2>	µV, dB

Mod FFT analyzer

:MKR1?	<NR2>	FM in Hz, AM in %, PM in rad
:MKR2?	<NR2>	FM in Hz, AM in %, PM in rad
:MKRDiff?	<NR2>, <NR2>	FM in Hz, dB AM in %, dB PM in rad, dB

CW or sideband analyzer

:MKR1?	<NR2>	dB
:MKR2?	<NR2>	dB
:MKRDiff?	<NR2>	dB

Digital power analyzer

:MKR1?	<NR2>	dB
:MKR2?	<NR2>	dB
:MKRDiff?	<NR2>	dB

Digital phase analyzer

:MKR1?	<NR2>	degrees
:MKR2?	<NR2>	degrees
:MKRDiff?	<NR2>	degrees

Instrument digital measurement commands

List of commands

:MEASDig
- :ALLBer?
- :ALLRber?
- :ALLTx?
- :BER1?
- :BER2?
- :Fer?
- :PHASEPeak?
- :PHASERms?
- :PRofile?
- :RBER1b?
- :RBER2?
- :RXreport?
- :TIming?
- :TXFreq?
- :TXPower?
- :VECTPeak?
- :VECTREs?
- :VECTRMs?

Measurement cycle

The measurement speed is affected by the measurement cycle mode, which determines the way in which measurements are returned over the remote control. Briefly, if the mode is ON, measurements are returned more quickly, whereas when it is OFF there is more certainty that measurements are current. For more information, refer to the description of the :MEASCYCLE command on page 5-112.

:MEASDig

:ALLBer?

Description : Returns the level and percentage complete for all BER measurements. Available in Digital DUPLEX and Systems modes when the mobile is in the relevant loopback mode (see Chapter 6, STEST:LOOPBK command). The number of samples can be set via :MEASAverages (page 5-105) or via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR2>, <NR2>, <NR2>
Class I BER, % settled, Class II BER, % settled

Example : `:MEASDig:ALLBer?`

Example response : `21.300, 100.0, 29.200, 98.6`
The measured BER for Class 1 is 21.3% and is 100% settled, and for Class II is 29.2% and 98.6% settled

:MEASDig

:ALLRber?

Description : Returns the the level and percentage complete for all RBER measurements. Available in Digital DUPLEX and Systems modes when the mobile is in the relevant loopback mode (see Chapter 6, STEST:LOOPBK command). The number of samples can be set via :MEASAverages (page 5-110) or via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>
Class I RBER, % settled, Class II RBER, % settled, FER, % settled

Example : `:MEASDig:ALLRber?`

Example response : `21.900,100.0,14.200,100.0,6.100,99.7`
The measured RBER for Class 1 is 21.9% and is 100% settled, for Class II is 14.2% and 100% settled, and the FER is 6.1% and is 99.7% settled

:MEASDig

Applicable to GSM, DCS and PCS systems. See below regarding TETRA Systems

:ALLTx?

Description : Returns all the digital Tx measurements. Available in Digital DUPLEX and Systems modes. The number of samples for Tx frequency error, RMS phase error and Peak phase error can be set via :MEASAverages (page 5-107, 109). The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, both the number of samples and the Tx measurement type can be set via *[digital systems]* in HELP SET-UP.

Response : <NR2>, <NR1>, <NR2>, <NR1>, <NR2>, <NR1>,
<CRD>, <NR2>, <NR1>, <NR2>, <NR1>
Tx frequency error, 0/1, RMS phase error, 0/1, Peak phase error, 0/1
Power profile PASS/FAIL, Timing, 0/1, Power, 0/1
Where 0 = not valid, 1 = valid

Example : `:MEASDig:ALLTx?`

Example response : `40.1,1,17.9,1,12.6,1,PASS,1.21,0,25.1,1`
The measured:-
Tx frequency error is 40.1 Hz and is valid
peak phase is 17.9° and is valid
RMS phase is 12.6° and is valid
The power profile has passed
The measured:-
timing is 1.21 bits and is not valid (yet)
power is 25.1 dBm and is valid

:MEASDig

Applicable to TETRA MOBILE, TETRA BASE and TETRA DIRECT systems. See above regarding GSM, DCS and PCS systems

:ALLTx?

Description : Returns all the digital Tx measurements.
Available in Digital DUPLEX and SYSTEMS modes.
The number of samples for Tx frequency error, Peak vector error RMS vector error, Residual Carrier and Power can be set via :MEASAverages (page 5-107, 109).
The Tx measurement type (worst case or average) for the above measurements can be set by :MEASAverages:Digtxtype (page 5-106).
Alternatively, both the number of samples and the Tx measurement type can be set via *[digital systems]* in HELP SET-UP.

Response : <NR2>, <NR1>, <CRD>, <NR2>, <NR1>, <NR2>, <NR1>, <NR2>, <NR1>, <NR2>, <NR1>, <NR2>, <NR1>,
Tx frequency error, 0/1, Peak vector error, 0/1, RMS vector error, 0/1, Residual carrier, 0/1 Power profile PASS/FAIL, Reserved, 0/1, Power, 0/1
Where 0 = not settled, 1 = settled

Example : `:MEASDig:ALLTx?`

Example response : `40.1,1,5.11,1,2.01,1,1.23,0,PASS,0.0,1,25.1,1`
The measured:-
Tx frequency error is 40.1 Hz and has settled
Peak vector error is 5.11% and has settled
RMS vector error is 2.01% and has settled
Residual carrier is 1.23% and has not settled
The power profile has passed
The reserved value is 0.0 and has settled
The measured:-
power is 25.1 dBm and has settled

:MEASDig

:BER1?

Description : Returns the value of Bit Error Rate Class I and the percentage settled. Available in Digital DUPLEX and Systems modes when the mobile is in the relevant loopback mode (see Chapter 6, STEST:LOOPBK command). The number of samples can be set via :MEASAverages:BER1 (page 5-105) or via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR2> Bit Error Rate in %, percentage settled in %

Example : `:MEASDig:BER1?`

Example response : `21.000,100.0`
The measured bit error rate is 21% and is 100% settled

:MEASDig

:BER2?

Description : Returns the value of Bit Error Rate Class II and the percentage settled. Available in Digital DUPLEX and Systems modes when the mobile is in the relevant loopback mode (see Chapter 6, STEST:LOOPBK command). The number of samples can be set via :MEASAverages:BER2 (page 5-106) or via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR2> Bit Error Rate in %, percentage settled in %

Example : `:MEASDig:BER2?`

Example response : `14.012,98.2`
The measured bit error rate is 14.012% and is 98.2% settled

:MEASDig

:Fer?

Description : Returns the value of Frame Erasure Rate and the percentage settled. Available in Digital DUPLEX and Systems modes when the mobile is in the relevant loopback mode (see Chapter 6, STEST:LOOPBK command). The number of samples can be set via :MEASAverages:FEr (page 5-107) or via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR2> Frame Erasure Rate in %, percentage settled in %

Example : `:MEASDig:Fer?`

Example response : `1.300, 12.1`
The measured Frame Erasure Rate is 1.3% and is 12.1% setlled

:MEASDig

:PHASEPeak?

Description : Returns the peak value of phase error measured over a number of bursts. The number of samples can be set via :MEASAverages:PHASEPeak (page 5-109). The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, both the number of samples and the Tx measurement type can be set via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR1> Peak phase error (degrees), 0 (not valid) or 1 (valid)

Example : `:MEASDig:PHASEPeak?`

Example response : `1.2,1`
The measured peak phase error is 1.2° and is valid

:MEASDig

:PHASERms?

Description : Returns the RMS value of phase error measured over a number of bursts. The number of samples can be set via :MEASAverages:PHASERms (page 5-109). The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, both the number of samples and the Tx measurement type can be set via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR1> RMS phase error (degrees), 0 (not valid) or 1 (valid)

Example : `:MEASDig:PHASERms?`

Example response : `4.1,1`
The measured RMS phase error is 4.1° and is valid

:MEASDig

:PRofile?

Description : Returns the profile status.

Response : <ARB_ASCII> PASS or FAIL

Example : `:MEASDig:PRofile?`

Example response : `PASS`
The power profile passed the defined limits

:MEASDig

:RBER1b?

Description : Returns the value of Residual Bit Error Rate Class Ib and the percentage settled. The number of samples can be set via :MEASAverages:RBER1b (page 5-110) or via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR2> RBER in %, percentage settled in %

Example : `:MEASDig:RBER1b?`

Example response : `14.012,98.2`
The measured residual bit error rate is 14.012% and is 98.2% settled

:MEASDig

:RBER2?

Description : Returns the value of Residual Bit Error Rate Class II and the percentage settled. The number of samples can be set via :MEASAverages:RBER2 (page 5-110) or via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR2> RBER in %, percentage settled in %

Example : `:MEASDig:RBER2?`

Example response : `14.012,98.2`
The measured residual bit error rate is 14.012% and is 98.2% settled

:MEASDig

:RXreport?

Description : Returns the RxLEV and RxQUAL as reported by the mobile on the SACCH, and whether the report is valid (e.g. invalid if no mobile Rx report is received within one second). Available in Systems mode only.

Response : <NR1>, <NR1>, <NR1> RxLEV, RxQUAL, 0 (not valid) or 1 (valid)

Example : `:MEASDig:RXreport?`

Example response : `8,5,1`
The measured values of RxLEV and RxQUAL are 8 (i.e. –103 dBm to –102 dBm) and 5 (i.e. 3.2 to 6.4% BER), and the report is valid.

:MEASDig

:TIming?

Description : Returns the value of mobile timing advance.

Response : <NR2>,<NR1> Timing in bits, 0 (not valid) or 1 (valid)

Example : `:MEASDig:TIming?`

Example response : `1.02,1`
The measured timing advance is 1.02 bits and is valid

:MEASDig

:TXFreq?

Description : Returns the Tx Frequency error measured over a number of bursts. The number of samples can be set via :MEASAverages:FReqerr (page 5-107). The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, both the number of samples and the Tx measurement type can be set via *[digital systems]* in HELP SET-UP.

Response : <NR2>, <NR1> Frequency error in Hz, 0 (not valid) or 1 (valid)

Example : `:MEASDig:TXFreq?`

Example response : `1.3,1`
The measured Tx Frequency error is 1.3 Hz and is valid

:MEASDig

:TXPower?

Description : Returns the value of digital Tx Power.

Response : <NR2>,<NR1> Power in dBm, 0 (not valid) or 1 (valid)

Example : `:MEASDig:TXPower?`

Example response : `-32.2,1`
The measured Tx Power is –32.2 dBm and is valid

:MEASDig

Applicable to TETRA MOBILE, TETRA BASE and TETRA DIRECT systems

:VECTPeak?

Description : Returns the peak value of vector error measured over a number of bursts.
Available in Digital DUPLEX and SYSTEMS modes.
The number of samples can be set via :MEASAverages:VECTPeak (page 5-109).
The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106).
Alternatively, both the number of samples and the Tx measurement type can be set via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR1> Peak vector error (%), 0 (not settled) or 1 (settled)

Example : `:MEASDig:VECTPeak?`

Example response : `1.2,1`
The measured peak vector error is 1.2% and has settled

:MEASDig

Applicable to TETRA MOBILE, TETRA BASE and TETRA DIRECT systems

:VECTREs?

Description : Returns the value of residual carrier measured over a number of bursts.
Available in Digital DUPLEX and SYSTEMS modes.
The number of samples can be set via :MEASAverages:VECTREs (page 5-109).
The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106).
Alternatively, both the number of samples and the Tx measurement type can be set via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR1> Residual Carrier (%), 0 (not settled) or 1 (settled)

Example : `:MEASDig:VECTREs?`

Example response : `4.1,1`
The measured residual carrier is 4.1% and has settled

:MEASDig

Applicable to TETRA MOBILE, TETRA BASE and TETRA DIRECT systems

:VECTRMs?

Description : Returns the RMS value of vector error measured over a number of bursts.
Available in Digital DUPLEX and SYSTEMS modes.
The number of samples can be set via :MEASAverages:VECTRms (page 5-109).
The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106).
Alternatively, both the number of samples and the Tx measurement type can be set via *[digital systems]* in HELP SET-UP.

Response : <NR2>,<NR1> RMS vector error (%), 0 (not settled) or 1 (sttled)

Example : `:MEASDig:VECTRms?`

Example response : `4.1,1`
The measured RMS vector error is 4.1% and has settled

Summary of digital measurement response formats

The measurements are returned as data, with no unit field, in the following formats:-

Measurement Type	Returned Format	Units/Representation
:ALLBer?	<NR2>,<NR2>,<NR2>,<NR2>	%, %, %, %
:ALLRber?	<NR2>,<NR2>,<NR2>,<NR2>, <NR2>,<NR2>	%, %, %, %, %, %
:ALLTx? (for GSM, DCS 1800 and PCS 1900 systems)	<NR2>,<NR1>,<NR2>,<NR1>, <NR2>,<NR1>, <CRD>, <NR2>,<NR1>,<NR2>,<NR1>	Hz, 0/1, degrees, 0/1, degrees, 0/1, PASS or FAIL, bits, 0/1, dBm, 0/1
:ALLTx? (for TETRA MOBILE, TETRA BASE and TETRA DIRECT systems)	<NR2>,<NR1>,<NR2>,<NR1>, <NR2>,<NR1>,<NR2>,<NR1>, <CRD>, <NR2>,<NR1>,<NR2>,<NR1>	Hz, 0/1, %, 0/1, %, 0/1, %, 0/1, PASS or FAIL, reserved, 0/1, dBm, 0/1
:BER1?	<NR2>,<NR2>	%, %
:BER2?	<NR2>,<NR2>	%, %
:Fer?	<NR2>,<NR2>	%, %
:PHASEPeak?	<NR2>,<NR1>	degrees, 0/1
:PHASERms?	<NR2>,<NR1>	degrees, 0/1
:PRofile?	<ARB_ASCII>	PASS or FAIL
:RBER1b?	<NR2>,<NR2>	%, %
:RBER2?	<NR2>,<NR2>	%, %
:RXreport?	<NR1>,<NR1>,<NR1>	none, none, 0/1
:TIming?	<NR2>,<NR1>	bits, 0/1
:TXFreq?	<NR2>,<NR1>	Hz, 0/1
:TXPower?	<NR2>,<NR1>	dBm, 0/1
:VECTPeak?	<NR2>,<NR1>	%, 0/1
:VECTREs?	<NR2>,<NR1>	%, 0/1
:VECTRMs?	<NR2>,<NR1>	%, 0/1

Memory and memory card commands

The memory and memory card commands enable you to carry out the store and recall functions described in the Operating Manual and its associated System Supplements.

List of commands

:MEMCard	:Title	:Name
:FOrmat	:Title?	:Name?
:FOrmat?	:Used?	:Protect
:FRee?	:MEMOry	:Protect?
:SIze?	:Catalog?	:Recall
:STatus?	:Delete	:Store

:MEMCard

:FOrmat

Description : Formats a memory card inserted in the test set memory card slot (the only choice is MI DOS format).

Parameters : <CPD> MI_DOS_FORMAT

Allowed suffices : Not applicable

Example : `:MEMCard:FOrmat MI_DOS_FORMAT`
Formats the memory card

:FOrmat?

Description : Returns the current format of the memory card.

Response : <CRD> UNFORMATTED
OLD_2965_FORMAT
MI_DOS_FORMAT

Example : `:MEMCard:FOrmat?`

Example response : `MEMC:FO MI_DOS_FORMAT`

:MEMCard

:FRee?

Description : Queries how much space is left on the memory card; query only.

Response : <NR1> Returns the number of bytes free.

Example : `:MEMCard:FRee?`

Example response : `MEMC:FR 379904`

:MEMCard

:SIze?

Description : Queries the size in bytes of the memory card; query only.

Response : <NR1> Returns the size in bytes

Example : `:MEMCard:SIze?`

Example response : `MEMC:SI 524288`

:MEMCard

:STatus?

Description : Queries the status of the memory card; query only.

Response : <CRD> BAT_FLAT
CARD_ABSENT
READ_AND_WRITE
READ_ONLY

Example : `:MEMCard:STatus?`

Example response : `MEMC:ST READ_AND_WRITE`

:MEMCard

:Title

Description : Names the memory card.

Parameters : <ARB_ASCII> Arbitrary ASCII string

Allowed suffices : Not applicable

Example : `:MEMCard:Title "MI Card 1"`
Sets the name of the memory card to MI Card 1

:Title?

Description : Returns the name of the memory card.

Response : <ARB_ASCII> Arbitrary ASCII string

Example : `:MEMCard:Title?`

Example response : `MEMC:T MI Card 1`

:MEMCard

:Used?

Description : Queries the number of memory card bytes that have been used; query only.

Response : <NR1>

Example : `:MEMCard:Used?`

Example response : `MEMC:U 136318`

:MEMOry

:Catalog?

Description : Queries all store location information, either internal or on a memory card; query only.

Response : <ARB_ASCII>

Example : `:MEMOry:Catalog?`

Example response :
```
01   Factory Default              Settings
02   Factory Default              Settings
03   Factory Default              Settings
.................(etc) .................
```

:MEMOry

:Delete

Description : Deletes a specified memory card store location (20-99). Action only.

Note: Internal store locations cannot be deleted.

Parameters : <NR1> Specifies the number of the memory card store location

Allowed suffices : Not applicable

Example : `:MEMOry:Delete 24`
Deletes memory card store location number 24

:MEMOry

:Name

Description : Names a specified store location. The location can be internal (store numbers 1-19) or on a memory card (store numbers 20-99).

Parameters : <NR1> n = 1-99
<ARB_ASCII> Arbitrary ASCII string

Allowed suffices : Not applicable

Example : `:MEMOry:Name 24, "GSM SETTINGS 1"`
Sets the name of store location 24 to GSM SETTINGS 1

:Name?

Description : Returns the name of the specified store location.

Parameters : <NR1> n = 1-99

Response : <ARB_ASCII> Arbitrary ASCII string

Example : `:MEMOry:Name? 24`

Example response : `MEMO:N GSM SETTINGS 1`

:MEMOry

:Protect

Description : Protects or unprotects a specified store location on a memory card only (store numbers 20-99).

Parameters : <NR1> n = 20-99
<CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:MEMOry:Protect 24, ON`
Protects memory card store location 24

:Protect?

Description : Returns the protection status of the specified store location.

Parameters : <NR1> n = 20-99

Response : <CRD> OFF, ON

Example : `:MEMOry:Protect? 24`

Example response : `MEMO:P ON`

:MEMOry

:Recall

Description : Recalls an instrument configuration from a specified store location, which may be internal (0-19) or on a memory card (20-99). Action only.

Parameters : <NR1> n = 0 to 99

Allowed suffices : Not applicable

Example : `:MEMOry:Recall 24`
Recalls the instrument configuration from memory card store location 24

:MEMOry

:Store

Description : Stores an instrument configuration to a specified store location, which may be internal (1-19) for instrument settings only, or on a memory card (20-99) for all types. Action only.

Parameters :

<NR1>	n = 1-99
<CPD>	RESULTS, SETTINGS, SYS_RESULTS, SYS_SETTINGS, BASIC (*Note:* SYS_RESULTS, SYS_SETTINGS *and* BASIC *are not applicable for* TETRA *systems)*
<ARB_ASCII>	Store location name

Allowed suffices : Not applicable

Example : `:MEMOry:Store 28, SYS_RESULTS, "Name"`
Stores System results to memory card store location 28 and gives it the name "Name" (which can be blank)

Instrument-specific commands

Note 1

Where sample responses are shown for Query commands, these are without headers. The instrument can be made to send various formats of response. This function is controlled by the :RESponse:Header command. See the alphabetic list below for details.

Command and Query examples are shown in the long form with the short form indicated by uppercase letters.

Note 2

Command sequences

In general, when setting up the instrument using remote commands, it is advisable to send the commands in an order that follows the sequence that you would use when operating the instrument locally. When a parameter setup command is sent that is not directly appropriate to the current setup, it is usually carried out in the background. For instance, setting up the spectrum analyzer parameters when in an AF TEST screen will not cause the spectrum analyzer to be displayed, but when it is, those parameters will be used. However, certain commands will return a device error when not in the correct mode. These tend to be associated with attempting to display operating screens or graphical data that is not appropriate to the current operating mode; for example, switching on the spectrum analyzer in AF TEST mode. If a device error is returned, and the command appears to be correct and in range, check that the instrument is in the correct mode.

List of commands

:ACcessories
- **:Dpowertype**
- **:Dpowertype?**
- **:LOGICn**
- **:LOGICn?**
- **:STATUSn**
- **:STATUSn?**

:AFGENErators

:AFGENErators?

:AFGENn
- **:Display**
- **:Freq**
- **:Freq?**
- **:Level**
- **:Level?**
- **:Power**
- **:Power?**
- **:Status**
- **:Status?**

:AFInput

:AFInput?

:AFMode

:AFMode?

AUdioana
- **:AFreflevel**
- **:AFreflevel?**

:AUdioana
- **:MARkers**
- **:MARkers?**
- **:MAXhold**
- **:MAXhold?**
- **:MKRn**
- **:MKRn?**
- **:PEakfind**
- **:Reffreq**
- **:Reffreq?**
- **:SIze**
- **:SIze?**
- **:SPan**
- **:SPan?**
- **:Trig**
- **:Trig?**
- **:Vrange**
- **:Vrange?**

:BArchart
- **:AFDistn**
- **:AFDistn?**
- **:AFLevel**
- **:AFLevel?**
- **:AFSInad**
- **:AFSInad?**
- **:AFSN**
- **:AFSN?**
- **:BER1**
- **:BER1?**
- **:BER2**
- **:BER2?**
- **:DIGPHPeak**
- **:DIGPHPeak?**
- **:DIGPHRms**
- **:DIGPHRms?**
- **:DIGPHType**
- **:DIGPHType?**
- **:FEr**
- **:FEr?**
- **:FReqerr**
- **:FReqerr?**
- **:FWdpower**
- **:FWdpower?**
- **:Maxhold**
  - **:AFDistn**
  - **:AFDistn?**
  - **:AFLevel**
  - **:AFLevel?**
  - **:AFSInad**
  - **:AFSInad?**

:AFSN
:AFSN?
:BER1
:BER1?
:BER2
:BER2?
:DIGPHPeak
:DIGPHPeak?
:DIGPHRms
:DIGPHRms?
:FEr
:FEr?
:FReqerr
:FReqerr?
:FWdpower
:FWdpower?
:RBER1b
:RBER1b?
:RBER2
:RBER2?
:REvpower
:REvpower?
:TXAmmod
:TXAmmod?
:TXBandpwr
:TXBandpwr?
:TXDIGpwr
:TXDIGpwr?
:TXDIStn
:TXDIStn?
:TXFmmod
:TXFmmod?
:TXLevel
:TXLevel?
:TXPmmod
:TXPmmod?
:TXSn
:TXSn?
:VECTPeak
:VECTPeak?
:VECTREs
:VECTREs?
:VECTRMs
:VECTRMs?
:VSwr
:VSwr?

:RBER1b
:RBER1b?
:RBER2
:RBER2?
:REvpower
:REvpower?
:TXAmmod
:TXAmmod?
:TXBandpwr
:TXBandpwr?
:TXDIGpwr
:TXDIGpwr?
:TXDIStn
:TXDIStn?
:TXFmmod
:TXFmmod?
:TXLevel
:TXLevel?
:TXPmmod
:TXPmmod?
:TXSn
:TXSn?
:VECTPeak
:VECTPeak?
:VECTREs
:VECTREs?
:VECTRMs
:VECTRMs?
:VSwr
:VSwr?

:BUrsttype
:BUrsttype?
:CAlibrate
:Audiofilters?
:DIGPowermeter?
:DIGSiggen?
:Multimeter?
:Powermeter?
:Specana?
:Warping?
:COMmerror?
CONdition
:Freqstd?
:COPy
:CWana
:MARkers
:MARkers?
:MAXhold
:MAXhold?
:MKRn
:MKRn?
:Peakfind
:REFFreq
:REFFreq?
:REFLevel
:REFLevel?
:SIze
:SIze?
:SPan
:SPan?

:Trig
:Trig?
:Vrange
:Vrange?

:DAFmode
:DAFmode?
:DATe
:DATe?
:DDxmode
:DDxmode?
:DEModtype
:DEModtype?
:DEVerror?
:DIsplay
:DIsplay?
:DUplexmode
:DUplexmode?
:EXecerror?
:Hwreset
:MEASAverages
:Amps
:Amps?
:BER1
:BER1?
:BER2
:BER2?
:Digtxtype
:Digtxtype?
:FEr
:FEr?
:FReqerr
:FReqerr?
:Ohms
:Ohms?
:POwer
:POwer?
:PHASEPeak
:PHASEPeak?
:PHASERms
:PHASERms?
:RBER1b
:RBER1b?
:RBER2
:RBER2?
:VECTPeak
:VECTPeak?
:VECTREs

:VECTREs?
:VECTRMs
:VECTRMs?
:MEASCycle
:MEASCycle?
:MODAna
:AMreflevel
:AMreflevel?
:Fmreflevel
:Fmreflevel?
:MARkers
:MARkers?
:MAXhold
:MAXhold?
:MKRn
:MKRn?
:PEakfind
:PMreflevel
:PMreflevel?
:Reffreq
:Reffreq?
:SIze
:SIze?
:SPan
:SPan?
:Trig
:Trig?
:Vrange
:Vrange?
:MODGENn
:Amdepth
:Amdepth?
:Display
:FMdevn
:FMdevn?
:FReq
:FReq?
:Pmdevn
:Pmdevn?
:Status
:Status?
:MODGENX
:Amdepth
:Amdepth?
:Display
:Fmdevn
:Fmdevn?
:Pmdevn
:Pmdevn?
:Status
:Status?
:MODInput
:MODInput?
:MODType
:MODType?
:MODUlation
:MODUlation?
:MUltimeter
:ACAmps
:ACAmps?
:ACVolts
:ACVolts?
:DCAmps
:DCAmps?
:DCVolts
:DCVolts?
:Maxhold
:Maxhold?
:Ohms
:Ohms?
:Type
:Type?
:PHaseana
:MArkers
:MArkers?
:MKRn
:MKRn?
:SCale
:SCale?
:SIze
:SIze?
:Trig
:Trig?
:PWrana
:MArkers
:MArkers?
:MKRn
:MKRn?
:MKRSYMn
:MKRSYMn?
:SCale
:SCale?
:SIze
:SIze?
:TRig
:TRig?
:TYpe
:TYpe?
:Qerror?
:RECAll
:RECEiver
:Autotune
:Autotune?
:Bursttype
:Bursttype?
:Channel
:Channel?
:Digautolev
: Digautolev?
:FREQ
:FREQ?
:FRESn
:FRESn?
:Ifband
:Ifband?
:Powerbw
:Powerbw?
:RESponse
:Header
:Header?
:RFGen
:Channel
:Channel?
:Freq
:Freq?
:Level
:Level?
:Mode
:Mode?
:Status
:Status?
:Topseamlevel
:Topseamlevel?
:RFInput
:RFInput?
:RFMode
:RFMode?
:RFPort
:RFPort?
:RXDIstn
:AFSlave
:AFSlave?
:AFUser
:AFUser?
:MODSlave
:MODSlave?
:MODUser
:MODUser?
:RXDType

:RXDType?
:RXEqtx
:RXFilt
:RXFilt?
:RXMode
:RXMode?
:SBandana
  :MARkers
  :MARkers?
  :MAXhold
  :MAXhold?
  :MKRn
  :MKRn?
  :Peakfind
  :Reflevel
  :Reflevel?
  :SIze
  :SIze?
  :SPan
  :SPan?
  :Trig
  :Trig?
  :Vrange
  :Vrange?
:SCope
  :AFrange
  :AFrange?
  :AMrange
  :AMrange?
  :Fmrange
  :Fmrange?
  :MArkers
  :MArkers?
  :MKRn
  :MKRn?
  :PMrange
  :PMrange?
  :POsItion
  :POsItion?
  :SIze
  :SIze?
  :SPan or :TBase
  :SPan? or :TBase?
  :TRig
  :TRig?
:SPecana
  :Filter
  :Filter?
  :MARkers
  :MARkers?
  :MAXhold
  :MAXhold?
  :MKRn
  :MKRn?
  :Peakfind
  :REFFreq
  :REFFreq?
  :REFLevel
  :REFLevel?
  :SIze
  :SIze?
  :SPan
  :SPan?
  :TRAck
  :TRAck?
  :TRIg
  :TRIg?
  :Vrange
  :Vrange?
:SQuelch
:SQuelch?
:SSb
  :Mode
  :Mode?
  :Rxfreq
  :Rxfreq?
  :Suppress
  :Suppress?
:STOre
:SWeepgen
  :Status
  :Status?
:TEstmode
:TEstmode?
:TIme
:TIme?
:TRackgen
  :MIxerIo
  :MIxerIo?
  :MOde
  :MOde?
  :Status
  :Status?
:TXDIstn
  :AFSlave
  :AFSlave?
  :AFUser
  :AFUser?
  :MODSlave
  :MODSlave?
  :MODUser
  :MODUser?
:TXDType
:TXDType?
:TXFilt
:TXFilt?
:TXMode
:TXMode?
:UNitmeas
  :Aflevel
  :Aflevel?
  :Demodlevel
  :Demodlevel?
:USeroptions
  :Afpad
  :Afpad?
  :Freqstd
  :Freqstd?
  :Genmode
  :Genmode?
  :Inlevoffset
  :Inlevoffset?
  :Language
  :Language?
  :PRINTPort
  :PRINTPort?
  :PRINTType
  :PRINTType?
  :PSoph_select
  :PSoph_select?
  :REMOTEIn
  :REMOTEIn?
  :REMOTEOut
  :REMOTEOut?
  :RFlevoffset
  :RFlevoffset?
  :Stdfreq
  :Stdfreq?
  :Termaf
  :Termaf?
:VECTDiag
  :Refresh
  :Refresh?
  :Size
  :Size?
  :TRig
  :TRig?
  :TYpe
  :TYpe?
:VOlume
:VOlume?

:ACcessories

:Dpowertype

Description : Controls how the directional power head accessory takes measurement of power. Carrier Wave or Peak Envelope Power.

Parameters : <CPD> CW, PEP

Allowed suffices : Not applicable

Example : `:ACcessories:Dpowertype PEP`
Select Peak Envelope Power as the measurement parameter

:Dpowertype?

Description : Returns the current setting of the directional power head accessory power measurement method.

Response : <CRD> CW, PEP

Example : `:ACcessories:Dpowertype?`

Example response : `AC:D PEP`

:ACcessories

:LOGIC*n*

Description : Controls the state of the accessory logic line *n* on the accessory port where $n = 0$ to 3.

Parameters : <CPD> LOW, HIGH

Allowed suffices : Not applicable

Example : `:ACcessories:LOGIC2 HIGH`
Set Logic Line 2 to HIGH

:LOGIC*n*?

Description : Returns the current setting of accessory logic line *n* on the accessory port where $n = 0$ to 3.

Response : <CRD> LOW, HIGH

Example : `:ACcessories:LOGIC2?`

Example response : `AC:LOGIC2 HIGH`

:ACcessories

:STATUS*n*

Description : Controls the state of the accessory status line *n* on the accessory port where *n* = 2 only, since status lines 1 and 3 cannot be set.

Parameters : <CPD> LOW, HIGH

Allowed suffices : Not applicable

Example : `:ACcessories:STATUS1 LOW`
Set Status 1 Line to LOW

:STATUS*n*?

Description : Returns the current setting of accessory status line on the accessory port where *n* = 1 to 3.

Response : <CRD> LOW, HIGH

Example : `:ACcessories:STATUS1?`

Example response : `AC:STATUS1 LOW`

:AFGENErators

Description : Controls the status of the AF generators. An OFF command overrules an ON command of an individual AF generator (:AFGEN1:STATUS ON).

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:AFGENErators ON`
Set the AF generator function to ON

:AFGENErators?

Description : Returns the current setting of the AF generators overall state.

Response : <CRD> OFF, ON

Example : `:AFGENErators?`

Example response : `AFGENE ON`

:AFGEN*n*

:Display

Description : This command displays the selected AF generator parameters if an appropriate screen is in use. *n* = 1 to 4. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:AFGEN2:Display`
Displays the parameters of AF generator number 2

:AFGEN*n*

:Freq

Description : Sets the frequency of AF generator *n* where *n* = 1 to 4

Parameters : <NRf>

Allowed suffices : `HZ, KHZ`

Example : `:AFGEN1:Freq 10.000KHZ`
Set AF generator 1 frequency to 10 kHz.

:Freq?

Description : Returns currently set frequency of the AF generator *n*

Response : <NR2> Units in Hz

Example : `:AFGEN1:Freq?`

Example response : `AFGEN1:F 10,000.0Hz`

:AFGEN*n*

:Level

Description : Sets the level of AF generator *n* where *n* = 1 to 4

Parameters : <NRf>

Allowed suffices : `UV, MV, V`

Example : `:AFGEN1:Level 100MV`
Set AF generator 1 level to 100 mV.

:Level?

Description : Returns currently set level of the AF generator *n*

Response : <NR2> Units in µV

Example : `:AFGEN1:Level?`

Example response : `AFGEN1:L 100000UV`

:AFGEN*n*

:Power

Description : Sets the power level of AF generator *n*, where *n* = 1 to 4, when the 600 Ω Interface Unit accessory is fitted

Parameters : <NRf>

Allowed suffices : DBM

Example : `:AFGEN1:Power 0DBM`
Sets the AF generator 1 output to provide 0 dBm into 600 Ω

:Power?

Description : Returns the power level of the specified AF generator *n* when the 600 Ω Interface Unit accessory is fitted

Response : <NR2>

Example : `:AFGEN1:Power?`

Example response : `AFGEN1:P 0DBM`

:AFGEN*n*

:Status

Description : Sets the status of AF generator *n* where *n* = 1 to 4. An OFF command overrules an ON AF generator command (:AFGENE ON).

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:AFGEN1:Status ON`
Set AF generator 1 status to ON.

:Status?

Description : Returns the status of the specified AF generator *n*

Response : <CRD> OFF, ON
Current selection

Example : `:AFGEN1:Status?`

Example response : `AFGEN1:S ON`

:AFInput

Description : Controls the coupling of the AF INPUT socket.

Parameters : <CPD> AC, DC

Allowed suffices : Not applicable

Example : `:AFInput AC`
Set the AF input coupling to AC

:AFInput?

Description : Returns the coupling of the AF INPUT socket.

Response : <CRD> AC, DC

Example : `:AFInput?`

Example response : `AFI AC`

:AFMode

Description : Sets the AF TEST screen when in AF TEST mode or the DIGITAL AF TEST screen when in DIGITAL AF TEST mode.

Parameters : <CPD> MENU
NORMAL
MULTIMETER
AF_TONES_SEQ_GEN
AF_TONES_SEQ_DEC
AF_TONES_DTMF
AF_TONES_DTMF_GEN
AF_TONES_DTMF_DEC

Note: the AF TONES *screens are not available in* DIGITAL AF TEST *mode*

Allowed suffices : Not applicable

Example : `:AFMode AF_TONES_SEQ_GEN`
Set the instrument into the AF TEST SEQUENTIAL TONES GENERATE screen when in an AF TEST mode.

:AFMode?

Description : Returns the current selection of the AF TEST screen or DIGITAL AF TEST screen.

Response : <CRD> MENU
NORMAL
MULTIMETER
AF_TONES_SEQ_GEN
AF_TONES_SEQ_DEC
AF_TONES_DTMF
AF_TONES_DTMF_GEN
AF_TONES_DTMF_DEC

Note: the AF TONES *screens are not available in* DIGITAL AF TEST *mode.*

Example : `:AFMode?`

Example response : `AFM AF_TONES_SEQ_GEN`

AUdioana

:AFreflevel

Description : Sets the AF reference level of the audio analyzer.

Parameters : <NRf>

Allowed suffices : `UV, MV, V`

Example : `:AUdioana:AFreflevel 100MV`
Set the AF reference level to 100 mV.

:AFreflevel?

Description : Returns the current setting of the AF reference level of the audio analyzer.

Response : <NR2> Units in µV

Example : `:AUdioana:AFreflevel?`

Example response : `AU:AF 100000UV`

:AUdioana

:MARkers

Description : Sets the status of the audio analyzer markers.

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:AUdioana:MARkers OFF`
Turn the audio analyzer markers OFF

:MARkers?

Description : Returns the current status of the audio analyzer markers.

Response : <CRD> OFF, ON

Example : `:AUdioana:MARkers?`

Example response : `AU:MAR OFF`

:AUdioana

:MAXhold

Description : Sets, releases, or resets, the Maximum Hold function on the audio analyzer

Parameters : <CPD> OFF
ON

Allowed suffices : Not applicable

Example : `:AUdioana:MAXhold ON`
Sets Maximum Hold function

:MAXhold?

Description : Returns the current setting of the Maximum Hold function

Response : <CRD> OFF
ON

Example : `:AUdioana:MAXhold?`

Example response : `AU:MAX ON`

:AUdioana

:MKR*n*

Description : Sets the frequency location of audio analyzer marker *n*, where *n* = 1 or 2.

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : :AUdioana:MKR1 25.5KHZ
Set audio analyzer marker 1 to 25.5 kHz

:MKR*n*?

Description : Returns the current location of audio analyzer marker *n*, where *n* = 1 or 2.

Response : <NR2> Units in Hz

Example : :AUdioana:MKR1?

Example response : AU:MKR1 25500HZ

:AUdioana

:PEakfind

Description : Causes the audio analyzer marker 1 to locate at the point of maximum signal level. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : :AUdioana:PEakfind
Locate audio analyzer marker 1 at the frequency where the peak audio level is currenly found.

:AUdioana

:Reffreq

Description : Sets the audio analyzer reference frequency.

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : :AUdioana:REFfreq 25.5KHZ
Set the audio analyzer reference frequency to 25.5 kHz.

:Reffreq?

Description : Returns the current setting of the audio analyzer reference frequency.

Response : <NR2> Units in Hz

Example : :AUdioana:REFfreq?

Example response : AU:REF 25500HZ

:AUdioana

:SIze

Description : Sets the view of the audio analyzer mode to normal or expanded.

Parameters : <CPD> NORMAL, EXPAND

Allowed suffices : Not applicable

Example : `:AUdioana:SIze NORMAL`
Set the audio analyzer display to normal view.

:SIze?

Description : Returns the current setting of the view of the audio analyzer mode.

Response : <CRD> NORMAL, EXPAND

Example : `:AUdioana:SIze?`

Example response : `AU:SI NORMAL`

:AUdioana

:SPan

Description : Sets the frequency span per division of the audio analyzer.

Parameters : <CPD> AS_5HZ
AS_10HZ
AS_25HZ
AS_50HZ
AS_100HZ
AS_250HZ
AS_500HZ
AS_1KHZ
AS_2_5KHZ
AS_5KHZ

Allowed suffices : Not applicable

Example : `:AUdioana:SPan AS_5HZ`
Set the audio analyzer span to 5 kHz/division

:SPan?

Description : Returns the current setting of the audio analyzer frequency span per division.

Response : <CRD> AS_5HZ
AS_10HZ
AS_25HZ
AS_50HZ
AS_100HZ
AS_250HZ
AS_500HZ
AS_1KHZ
AS_2_5KHZ
AS_5KHZ

Example : `:AUdioana:SPan?`

Example response : `AU:SP AS_5HZ`

:AUdioana

:Trig

Description : Controls the audio analyzer trigger mode.

Parameters : <CPD> SINGLE, REPEAT

Allowed suffices : Not applicable

Example : `:AUdioana:Trig SINGLE`
Set the audio analyzer trigger mode to single sweep.

:Trig?

Description : Returns the current setting of the audio analyzer trigger mode.

Response : <CRD> SINGLE, REPEAT

Example : `:AUdioana:Trig?`

Example response : `AU:T SINGLE`

:AUdioana

:Vrange

Description : Sets the vertical span/division of the audio analyzer.

Parameters : <NRf>

Allowed suffices : `DB`

Example : `:AUdioana:Vrange 2DB`
Sets the vertical scaling to 2 dB per division

:Vrange?

Description : Returns current setting of vertical range/division of audio analyzer.

Response : <NR2> Units in dB

Example : `:AUdioana:Vrange?`

Example response : `AU:V 2DB`

:BArchart

:AFDistn

Description : Sets the AF Distortion bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
AD_1PC
AD_3PC
AD_10PC
AD_30PC

Allowed suffices : Not applicable

Example : `:BArchart:AFDistn AD_1PC`
Set the AF Distortion bar chart range to 1%

:AFDistn?

Description : Returns the current setting of the AF Distortion bar chart range.

Response : <CRD> AD_1PC
AD_3PC
AD_10PC
AD_30PC

Example : `:BArchart:AFDistn?`

Example response : `B:AFD AD_1PC`

:BArchart

:AFLevel

Description : Sets the AF Level bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
AL_1MV
AL_3MV
AL_10MV
AL_30MV
AL_100MV
AL_300MV
AL_1V
AL_3V
AL_10V
AL_30V

Allowed suffices : Not applicable

Example : `:BArchart:AFLevel AL_1MV`
Set the AF Level bar chart range to 1 mV

:AFLevel?

Description : Returns the current setting of the AF Level bar chart range.

Response : <CRD> AL_1MV
AL_3MV
AL_10MV
AL_30MV
AL_100MV
AL_300MV
AL_1V
AL_3V
AL_10V
AL_30V

Current selection

Example : `:BArchart:AFLevel?`

Example response : `B:AFL AL_1V`

:BArchart

:AFSInad

Description : Sets the AF SINAD bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
ASI_18DB
ASI_50DB

Allowed suffices : Not applicable

Example : `:BArchart:AFSInad ASI_18DB`
Set the AF SINAD bar chart range to 18 dB

:AFSInad?

Description : Returns the current setting of the AF SINAD bar chart range.

Response : <CRD> ASI_18DB
ASI_50DB

Example : `:BArchart:AFSInad?`

Example response : `B:AFSI ASI_18DB`

:BArchart

:AFSN

Description : Sets the Audio Signal/Noise bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
ASN_30DB
ASN_100DB

Allowed suffices : Not applicable

Example : :BArchart:`AFSN ASN_30DB`
Set the Audio Signal/Noise bar chart range to 30 dB

:AFSN?

Description : Returns the current setting of the Audio Signal/Noise bar chart range.

Response : <CRD> ASN_30DB
ASN_100DB

Example : `:BArchart:AFSN?`

Example response : `B:AFSN ASN_30DB`

:BArchart

:BER1

Description : Sets the bar chart range for Bit Error Rate Class I to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Allowed suffices : Not applicable

Example : `:BArchart:BER1 BER_3PC`
Set the bar chart for Bit Error Rate Class I to 3%

:BER1?

Description : Returns the current setting of the Bit Error Rate Class I bar chart range.

Response : <CRD> BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Example : `:BArchart:BER1?`

Example response : `B:BER1 BER_3PC`

:BArchart

:BER2

Description : Sets the bar chart range for Bit Error Rate Class 2 to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Allowed suffices : Not applicable

Example : `:BArchart:BER2 BER_3PC`
Set the bar chart for Bit Error Rate Class 2 to 3%

:BER2?

Description : Returns the current setting of the Bit Error Rate Class 2 bar chart range.

Response : <CRD> BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Example : `:BArchart:BER2?`

Example response : `B:BER2 BER_3PC`

:BArchart

:DIGPHPeak

Description : Sets the Digital Peak phase bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
PH_1DEG
PH_3DEG
PH_10DEG
PH_30DEG

Allowed suffices : Not applicable

Example : `:BArchart:DIGPHPeak PH_3DEG`
Set the digital peak phase bar chart range to 3°

:DIGPHPeak?

Description : Returns the current setting of the digital peak phase bar chart range.

Response : <CRD> PH_1DEG
PH_3DEG
PH_10DEG
PH_30DEG

Example : `:BArchart:DIGPHPeak?`

Example response : `B:DIGPHP PH_3DEG`

:BArchart

:DIGPHRms

Description : Sets the digital RMS phase bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
PH_1DEG
PH_3DEG
PH_10DEG
PH_30DEG

Allowed suffices : Not applicable

Example : `:BArchart:DIGPHRms PH_3DEC`
Set the digital RMS phase bar chart range to 3°

:DIGPHRms?

Description : Returns the current setting of the digital RMS phase bar chart range.

Response : <CRD> PH_1DEG
PH_3DEG
PH_10DEG
PH_30DEG

Example : `:BArchart:DIGPHRms?`

Example response : `B:DIGPHR PH_3DEG`

:BArchart

:DIGPHType

Description : Sets the digital phase bar chart type.

Parameters : <CPD> RMS
PEAK

Allowed suffices : Not applicable

Example : `:BArchart:DIGPHType RMS`
Set the phase bar chart to RMS

:DIGPHType?

Description : Returns the current setting of the digital phase bar chart type.

Response : <CRD> RMS
PEAK

Example : `:BArchart:DIGPHType?`

Example response : `B:DIGPHT RMS`

:BArchart

:FEr

Description : Sets the bar chart range for Frame erasure rate to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Allowed suffices : Not applicable

Example : `:BArchart:FEr BER_3PC`
Set the bar chart for Frame erasure rate to 3%

:FEr?

Description : Returns the current setting of the Frame Erasure rate bar chart range.

Response : <CRD> BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Example : `:BArchart:FEr?`

Example response : `B:FE BER_3PC`

:BArchart

:FReqerr

Description : Sets the Tx Frequency error bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
TX_100HZ
TX_300HZ
TX_1KHZ
TX_3KHZ
TX_10KHZ
TX_30KHZ

Allowed suffices : Not applicable

Example : `:BArchart:FReqerr TX_10KHZ`
Set the Tx Frequency error bar chart range to 10 kHz

:FReqerr?

Description : Returns the current setting of the Tx Frequency error bar chart range.

Response : <CRD> TX_100HZ
TX_300HZ
TX_1KHZ
TX_3KHZ
TX_10KHZ
TX_30KHZ

Example : `:BArchart:FReqerr?`

Example response : `B:FR TX_10KHZ`

:BArchart

:FWdpower

Description : Sets the Forward Power bar chart range of the directional power head accessory to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
DP_100MW
DP_1W
DP_10W
DP_100W
DP_500W

Allowed suffices : Not applicable

Example : `:BArchart:FWdpower DP_10W`
Set the directional power head accessory Forward Power bar chart range to 10 W

:FWdpower?

Description : Returns the current setting of the Forward Power bar chart range of the directional power head accessory.

Response : <CPD> DP_100MW
DP_1W
DP_10W
DP_100W
DP_500W

Example : `:BArchart:FWdpower?`

Example response : `B:FW DP_10W`

:BArchart

:Maxhold

:AFDistn

Description : Sets the Maximum Hold state for the AF distortion bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:AFDistn ON`
Sets the AF Distortion bar chart to Maximum Hold

:Maxhold

:AFDistn?

Description : Returns current value for AF Distortion bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:AFDistn?`

Example response : `B:M:AFD ON`

:BArchart

:Maxhold

:AFLevel

Description : Sets the Maximum Hold state for the AF Level bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:AFLevel ON`
Sets the AF Level bar chart to Maximum Hold

:Maxhold

:AFLevel?

Description : Returns current value for AF Level bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:AFLevel?`

Example response : `B:M:AFL ON`

:BArchart

:Maxhold

:AFSInad

Description : Sets the Maximum Hold state for the AF SINAD bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:AFSInad ON`
Sets the AF SINAD bar chart to Maximum Hold

:Maxhold

:AFSInad?

Description : Returns current value for AF SINAD bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:AFSInad?`

Example response : `B:M:AFSI ON`

:BArchart

:Maxhold

:AFSN

Description : Sets the Maximum Hold state for the AF S/N bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:AFSN ON`
Sets the AF S/N bar chart to Maximum Hold

:Maxhold

:AFSN?

Description : Returns current value for AF S/N bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:AFSN?`

Example response : `B:M:AFSN ON`

:BArchart

:Maxhold

:BER1

Description : Sets the Maximum Hold state for the Bit Error Rate I bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:BER1 ON`
Sets the Bit Error Rate I bar chart to Maximum Hold

:Maxhold

:BER1?

Description : Returns current value for Bit Error Rate I bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:BER1?`

Example response : `B:M:BER1 ON`

:BArchart

:Maxhold

:BER2

Description : Sets the Maximum Hold state for the Bit Error Rate II bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:BER2 ON`
Sets the Bit Error Rate II bar chart to Maximum Hold

:Maxhold

:BER2?

Description : Returns current value for Bit Error Rate II bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:BER2?`

Example response : `B:M:BER2 ON`

:BArchart

:Maxhold

:DIGPHPeak

Description : Sets the Maximum Hold state for the Digital Peak Phase Error bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:DIGPHPeak ON`
Sets the Digital Peak Phase Error bar chart to Maximum Hold

:Maxhold

:DIGPHPeak?

Description : Returns current value for Digital Peak Phase Error bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:DIGPHPeak?`

Example response : `B:M:DIGPHP ON`

:BArchart

:Maxhold

:DIGPHRms

Description : Sets the Maximum Hold state for the Digital RMS Phase Error bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:DIGPHRms ON`
Sets the Digital RMS Phase Error bar chart to Maximum Hold

:Maxhold

:DIGPHRms?

Description : Returns current value for Digital RMS Phase Error bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:DIGPHRms?`

Example response : `B:M:DIGPHR ON`

:BArchart

:Maxhold

:FEr

Description : Sets the Maximum Hold state for the Frame Erasure bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:FEr ON`
Sets the Frame Erasure bar chart to Maximum Hold

:Maxhold

:FEr?

Description : Returns current value for Frame Erasure bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:FEr?`

Example response : `B:M:FE ON`

:BArchart

:Maxhold

:FReqerr

Description : Sets the Maximum Hold state for the Frequency Error bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:FReqerr ON`
Sets the Frequency Error bar chart to Maximum Hold

:Maxhold

:FReqerr?

Description : Returns current value for Frequency Error bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:FReqerr?`

Example response : `B:M:FR ON`

:BArchart

:Maxhold

:FWdpower

Description : Sets the Maximum Hold state for the Forward Power bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:FWdpower ON`
Sets the Forward Power bar chart to Maximum Hold

:Maxhold

:FWdpower?

Description : Returns current value for Forward Power bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:FWdpower?`

Example response : `B:M:FW ON`

:BArchart

:Maxhold

:RBER1b

Description : Sets the Maximum Hold state for the Residual Bit Error Rate Ib bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:RBER1b ON`
Sets the Residual Bit Error Rate Ib bar chart to Maximum Hold

:Maxhold

:RBER1b?

Description : Returns current value for Residual Bit Error Rate Ib bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:RBER1b?`

Example response : `B:M:RBER1 ON`

:BArchart

:Maxhold

:RBER2

Description : Sets the Maximum Hold state for the Residual Bit Error Rate II bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:RBER2 ON`
Sets the Residual Bit Error Rate II bar chart to Maximum Hold

:Maxhold

:RBER2?

Description : Returns current value for Residual Bit Error Rate II bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:RBER2?`

Example response : `B:M:RBER2 ON`

:BArchart

:Maxhold

:REvpower

Description : Sets the Maximum Hold state for the Reverse Power bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:REvpower ON`
Sets the Reverse Power bar chart to Maximum Hold

:Maxhold

:REvpower?

Description : Returns current value for Reverse Power bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:REvpower?`

Example response : `B:M:RE ON`

:BArchart

:Maxhold

:TXAmmod

Description : Sets the Maximum Hold state for the Tx AM bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:TXAmmod ON`
Sets the Tx AM bar chart to Maximum Hold

:Maxhold

:TXAmmod?

Description : Returns current value for Tx AM bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:TXAmmod?`

Example response : `B:M:TXA ON`

:BArchart

:Maxhold

:TXBandpwr

Description : Sets the Maximum Hold state for the Tx In-Band Power Level bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:TXBandpwr ON`
Sets the Tx In-Band Power Level bar chart to Maximum Hold

:Maxhold

:TXBandpwr?

Description : Returns current value for Tx In-Band Power Level bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:TXBandpwr?`

Example response : `B:M:TXB ON`

:BArchart

:Maxhold

:TXDIGpwr

Description : Sets the Maximum Hold state for the Tx Digital Power Level bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:TXDIGpwr ON`
Sets the Tx Digital Power Level bar chart to Maximum Hold

:Maxhold

:TXDIGpwr?

Description : Returns current value for Tx Digital Power Level bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:TXDIGpwr?`

Example response : `B:M:TXDIG ON`

:BArchart

:Maxhold

:TXDIStn

Description : Sets the Maximum Hold state for the Tx Distortion bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:TXDIStn ON`
Sets the Tx Distortion bar chart to Maximum Hold

:Maxhold

:TXDIStn?

Description : Returns current value for Tx Distortion bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:TXDIStn?`

Example response : `B:M:TXDIS ON`

:BArchart

:Maxhold

:TXFmmod

Description : Sets the Maximum Hold state for the Tx FM bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:TXFmmod ON`
Sets the Tx FM bar chart to Maximum Hold

:Maxhold

:TXFmmod?

Description : Returns current value for Tx FM bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:TXFmmod?`

Example response : `B:M:TXF ON`

:BArchart

:Maxhold

:TXLevel

Description : Sets the Maximum Hold state for the Tx Level bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:TXLevel ON`
Sets the Tx Level bar chart to Maximum Hold

:Maxhold

:TXLevel?

Description : Returns current value for Tx Level bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:TXLevel?`

Example response : `B:M:TXL ON`

:BArchart

:Maxhold

:TXPmmod

Description : Sets the Maximum Hold state for the Tx Phase Modulation bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:TXPmmod ON`
Sets the Tx Phase Modulation bar chart to Maximum Hold

:Maxhold

:TXPmmod?

Description : Returns current value for Tx Phase Modulation bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:TXPmmod?`

Example response : `B:M:TXP ON`

:BArchart

:Maxhold

:TXSn

Description : Sets the Maximum Hold state for the Tx S/N bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:TXSn ON`
Sets the Tx S/N bar chart to Maximum Hold

:Maxhold

:TXSn?

Description : Returns current value for Tx S/N bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:TXSn?`

Example response : `B:M:TXS ON`

:BArchart

:Maxhold

:VECTPeak

Description : Sets the Maximum Hold state for the Peak Vector error bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:VECTPeak ON`
Sets the peak vector error bar chart to Maximum Hold

:Maxhold

:VECTPeak?

Description : Returns current value for Peak Vector error bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:VECTPeak?`

Example response : `BA:M:VECTP ON`

:BArchart

:Maxhold

:VECTREs

Description : Sets the Maximum Hold state for the Residual Carrier bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:VECTREs ON`
Sets the residual carrier bar chart to Maximum Hold

:Maxhold

:VECTREs?

Description : Returns current value for Residual Carrier bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:VECTREs?`

Example response : `BA:M:VECTRE ON`

:BArchart

:Maxhold

:VECTRMs

Description : Sets the Maximum Hold state for the RMS Vector error bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:VECTRMS ON`
Sets the RMS vector error bar chart to Maximum Hold

:Maxhold

:VECTRMs?

Description : Returns current value for RMS vector error bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:VECTRMs?`

Example response : `BA:M:VECTRM ON`

:BArchart

:Maxhold

:VSwr

Description : Sets the Maximum Hold state for the VSWR bar chart.

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:BArchart:Maxhold:VSwr ON`
Sets the VSWR bar chart to Maximum Hold

:Maxhold

:VSwr?

Description : Returns current value for VSWR bar chart Maximum Hold state.

Response : <CRD> OFF
ON

Example : `:BArchart:Maxhold:VSwr?`

Example response : `B:M:V ON`

:BArchart

:RBER1b

Description : Sets the bar chart range for Residual Bit Error Rate Class Ib to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Allowed suffices : Not applicable

Example : `:BArchart:RBER1b BER_3PC`
Set the bar chart for Residual Bit Error Rate Class Ib to 3%

:RBER1b?

Description : Returns the current setting of the Residual Bit Error Rate Class Ib bar chart range.

Response : <CRD> BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Example : `:BArchart:RBER1b?`

Example response : `B:RBER1 BER_3PC`

:BArchart

:RBER2

Description : Sets the bar chart range for Residual Bit Error Rate Class 2 to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Allowed suffices : Not applicable

Example : `:BArchart:RBER2 BER_3PC`
Set the bar chart for Residual Bit Error Rate Class 2 to 3%

:RBER2?

Description : Returns the current setting of the Residual Bit Error Rate Class 2 bar chart range.

Response : <CRD> BER_0_1PC
BER_0_3PC
BER_1PC
BER_3PC
BER_10PC
BER_30PC
BER_100PC

Example : `:BArchart:RBER2?`

Example response : `B:RBER2 BER_3PC`

:BArchart

:REvpower

Description : Sets the reverse power bar chart range of the directional power head accessory to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
DP_100MW
DP_1W
DP_10W
DP_100W
DP_500W

Allowed suffices : Not applicable

Example : `:BArchart:REvpower DP_100MW`
Set the directional power head accessory reverse power bar chart range to 100 mW

:REvpower?

Description : Returns the current setting of the reverse power bar chart range of the directional power head accessory bar chart range.

Response : <CPD> DP_100MW
DP_1W
DP_10W
DP_100W
DP_500W

Example : `:BArchart:REvpower?`

Example response : `B:RE DP_100MW`

:BArchart

:TXAmmod

Description : Sets the Tx AM bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
AML_10PC
AML_30PC
AML_100PC

Allowed suffices : Not applicable

Example : `:BArchart:TXAmmod AML_10PC`
Set the Tx AM bar chart range to 10%

:TXAmmod?

Description : Returns the current setting of the Tx modulation bar chart range.

Response : <CRD> AML_10PC
AML_30PC
AML_100PC

Example : `:BArchart:TXAmmod?`

Example response : `B:TXA AML_10PC.`

:BArchart

:TXBandpwr

Description : Sets the Tx in-band power level bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
MINUS_90DBM
MINUS_80DBM
MINUS_70DBM
MINUS_60DBM
MINUS_50DBM
MINUS_40DBM
MINUS_30DBM
MINUS_20DBM
MINUS_10DBM
PLUS_00DBM
PLUS_10DBM
PLUS_20DBM
PLUS_30DBM
PLUS_40DBM
PLUS_50DBM

Allowed suffices : Not applicable

Example : `:BArchart:TXBandpwr MINUS_90DBM`
Set the Tx in-band power level bar chart range to –90 dBm.

:TXBandpwr?

Description : Returns the current setting of the Tx in-band power level bar chart range.

Response : <CRD> MINUS_90DBM
MINUS_80DBM
MINUS_70DBM
MINUS_60DBM
MINUS_50DBM
MINUS_40DBM
MINUS_30DBM
MINUS_20DBM
MINUS_10DBM
PLUS_00DBM
PLUS_10DBM
PLUS_20DBM
PLUS_30DBM
PLUS_40DBM
PLUS_50DBM

Example : `:BArchart:TXBandpwr?`

Example response : `B:TXB MINUS_90DBM.`

:BArchart

:TXDIGpwr

Description : Sets the Digital Power bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
MINUS_90DBM
MINUS_80D BM
MINUS_70DBM
MINUS_60DBM
MINUS_50DBM
MINUS_40DBM
MINUS_30DBM
MINUS_20DBM
MINUS_10DBM
PLUS_00DBM
PLUS_10DBM
PLUS_20DBM
PLUS_30DBM
PLUS_40DBM
PLUS_50DBM

Allowed suffices : Not applicable

Example : `:BArchart:TXDIGpwr PLUS_20DBM`
Set the digital power bar chart range to 20 dBm

:TXDIGpwr?

Description : Returns the current setting of the bar chart range.

Response : <CRD> MINUS_90DBM
MINUS_80D BM
MINUS_70DBM
MINUS_60DBM
MINUS_50DBM
MINUS_40DBM
MINUS_30DBM
MINUS_20DBM
MINUS_10DBM
PLUS_00DBM
PLUS_10DBM
PLUS_20DBM
PLUS_30DBM
PLUS_40DBM
PLUS_50DBM

Example : `:BArchart:TXDIGpwr?`

Example response : `B:TXDIG PLUS_20DBM`

:BArchart

:TXDIStn

Description : Sets the Tx Distortion bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
MD_1PC
MD_3PC
MD_10PC
MD_30PC

Allowed suffices : Not applicable

Example : `:BArchart:TXDIStn MD_1PC`
Set the Tx Distortion bar chart range to 1%.

:TXDIStn?

Description : Returns the current setting of the Tx Distortion bar chart range.

Response : <CRD> MD_1PC
MD_3PC
MD_10PC
MD_30PC

Example : `:BArchart:TXDIStn?`

Example response : `B:TXDIS MD_1PC`

:BArchart

:TXFmmod

Description : Sets the Tx FM bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
FML_1KHZ
FML_3KHZ
FML_10KHZ
FML_30KHZ
FML_100KHZ
FML_300KHZ

Allowed suffices : Not applicable

Example : `:BArchart:TXFmmod FML_1KHZ`
Set the Tx FM bar chart range to 1 kHz.

:TXFmmod?

Description : Returns the current setting of the Tx FM bar chart range.

Response : <CRD> FML_1KHZ
FML_3KHZ
FML_10KHZ
FML_30KHZ
FML_100KHZ
FML_300KHZ

Example : `:BArchart:TXFmmod?`

Example response : `B:TXF FML_1KHZ`

:BArchart

:TXLevel

Description : Sets the TX Broad-Band Power Level bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
PWR_300UW
PWR_1MW
PWR_3MW
PWR_10MW
PWR_30MW
PWR_100MW
PWR_300MW
PWR_1W
PWR_3W
PWR_10W
PWR_30W
PWR_100W
PWR_300W

Allowed suffices : Not applicable

Example : `:BArchart:TXLevel PWR_300UW`
Set the Tx RF level bar chart range to 300 μW.

:TXLevel?

Description : Returns the current setting of the Tx Broad-Band Power Level bar chart range.

Response : <CRD> PWR_300UW
PWR_1MW
PWR_3MW
PWR_10MW
PWR_30MW
PWR_100MW
PWR_300MW
PWR_1W
PWR_3W
PWR_10W
PWR_30W
PWR_100W
PWR_300W

Example : `:BArchart:TXLevel?`

Example response : `B:TXL PWR_300UW`

:BArchart

:TXPmmod

Description : Sets the Tx PM bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
PML_1RAD
PML_3RAD
PML_10RAD
PML_30RAD

Allowed suffices : Not applicable

Example : `:BArchart:TXPmmod PML_1RAD`
Set the Tx Phase Modulation bar chart range to 1 radian.

:TXPmmod?

Description : Returns the current setting of the TX Phase Modulation bar chart range.

Response : <CRD> PML_1RAD
PML_3RAD
PML_10RAD
PML_30RAD

Example : `:BArchart:TXPmmod?`

Example response : `B:TXP PML_1RAD`

:BArchart

:TXSn

Description : Sets the TX S/N bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
MSN_30DB
MSN_100DB

Allowed suffices : Not applicable

Example : `:BArchart:TXSn MSN_30DB`
Set the Tx S/N bar chart range to 30 dB.

:TXSn?

Description : Returns the current setting of the Tx S/N bar chart range.

Response : <CRD> MSN_30DB
MSN_100DB

Example : `:BArchart:TXSn?`

Example response : `B:TXS MSN_30DB`

:BArchart

:VECTPeak

Description : Sets the Peak Vector error bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
VE_1PC
VE_3PC
VE_10PC
VE_30PC
VE_100PC

Allowed suffices : Not applicable

Example : `:BArchart:VECTPeak VE_3PC`
Set the peak vector error bar chart range to 3%

:VECTPeak?

Description : Returns the current setting of the Peak Vector error bar chart range.

Response : <CRD> VE_1PC
VE_3PC
VE_10PC
VE_30PC
VE_100PC

Example : `:BArchart:VECTPeak?`

Example response : `BA:VECTP VE_3PC`

:BArchart

:VECTREs

Description : Sets the Residual Carrier bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
VE_1PC
VE_3PC
VE_10PC
VE_30PC
VE_100PC

Allowed suffices : Not applicable

Example : `:BArchart:VECTREs VE_3PC`
Set the residual carrier bar chart range to 3%

:VECTREs?

Description : Returns the current setting of the Residual Carrier bar chart range.

Response : <CRD> VE_1PC
VE_3PC
VE_10PC
VE_30PC
VE_100PC

Example : `:BArchart:VECTREs?`

Example response : `BA:VECTRE VE_3PC`

:BArchart

:VECTRMs

Description : Sets the RMS Vector error bar chart range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
VE_1PC
VE_3PC
VE_10PC
VE_30PC
VE_100PC

Allowed suffices : Not applicable

Example : `:BArchart:VECTRMs VE_3PC`
Set the RMS vector error bar chart range to 3%

:VECTRMs?

Description : Returns the current setting of the RMS Vector error bar chart range.

Response : <CRD> VE_1PC
VE_3PC
VE_10PC
VE_30PC
VE_100PC

Example : `:BArchart:VECTRMs?`

Example response : `BA:VECTRM VE_3PC`

:BArchart

:VECTType

Description : Sets the vector error bar chart type.

Parameters : <CPD> RMS
PEAK
RESIDUAL

Allowed suffices : Not applicable

Example : `:BArchart:VECTType RMS`
Set the vector error bar chart to RMS

:VECTType?

Description : Returns the current setting of the vector error bar chart type.

Response : <CRD> RMS
PEAK
RESIDUAL

Example : `:BArchart:VECTType?`

Example response : `B:VECTT RMS`

:BArchart

:VSwr

Description : Sets the VSWR bar chart range of the directional power head accessory to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
VSWR_3
VSWR_10
VSWR_30
VSWR_100

Allowed suffices : Not applicable

Example : `:BArchart:VSwr VSWR_3`
Set the directional power head accessory VSWR bar chart range to 3:1

:VSwr?

Description : Returns the current setting of the VSWR bar chart range of the directional power head accessory.

Response : <CRD> VSWR_3
VSWR_10
VSWR_30
VSWR_100

Example : `:BArchart:VSwr?`

Example response : `B:V VSWR_3`

:BUrsttype

Description : Sets the type of TETRA burst to be measured

Parameters : <CPD> CONTROL, NORMAL or NORMAL_CONTROL
for TETRA MOBILE system

or SYNC, NORMAL or NORMAL_SYNC
for TETRA BASE and TETRA DIRECT systems

Note: NORMAL_CONTROL *and* NORMAL_SYNC *are not valid in the Power Analyzer display.*

Not applicable

Example : `:BUrsttype NORMAL`
Sets the type of burst to be measured to normal bursts

:BUrsttype?

Description : Returns the current burst type to be measured

Response : <CPD> CONTROL, NORMAL or NORMAL_CONTROL
for TETRA MOBILE system

or SYNC, NORMAL or NORMAL_SYNC
for TETRA BASE and TETRA DIRECT systems

Note: NORMAL_CONTROL *and* NORMAL_SYNC *are not valid in the Power Analyzer display*

Example : `:BUrsttype?`

Example response : `BU NORMAL`

:CAlibrate

:Audiofilters?

Description : Runs the audio filters automatic calibration; query only.

Response : <NR1> Returns a 0 when calibration is complete

Example : `:CAlibrate:Audiofilters?`

Example response : `0`

:CAlibrate

:DIGPowermeter?

Description : Runs the digital power meter automatic calibration; query only.

Response : <NR1> Returns a 0 when calibration is complete

Example : `:CAlibrate:DIGPowermeter?`

Example response : `0`

:CAlibrate

:DIGSiggen?

Description : Runs the digital signal generator automatic calibration; query only.
Response : <NR1> Returns a 0 when calibration is complete
Example : `:CAlibrate:DIGSiggen?`
Example response : `0`

:CAlibrate

:Multimeter?

Description : Runs the multimeter automatic calibration; query only.
Response : <NR1> Returns a 0 when calibration is complete
Example : `:CAlibrate:Multimeter?`
Example response : `0`

:CAlibrate

:Powermeter?

Description : Runs the power meter automatic calibration; query only.
Response : <NR1> Returns a 0 when calibration is complete
Example : `:CAlibrate:Powermeter?`
Example response : `0`

:CAlibrate

:Specana?

Description : Runs the Spectrum analyzer automatic calibration; query only.
Response : <NR1> Returns a 0 when calibration is complete
Example : `:CAlibrate:Specana?`
Example response : `0`

:CAlibrate

:Warping?

Description : Runs the warping automatic calibration; query only.
Response : <NR1> Returns a 0 when calibration is complete
Example : `:CAlibrate:Warping?`
Example response : `0`

:COMmerror?

Description : Returns the last command error generated by the remote parser; query only.

Response : <NR1>

0 'No Error'.

1 'Illegal Star Command'.

2 'Sub Command Not Allowed'.

3 'No Header Match Found'.
The command received was not one recognised by the parser.

4 'Header Text Is Not Unique'.
An abbreviated command mnemonic was received which was too short to uniquely identify one command, e.g. :MEAS:AFFreq?.

5 'Action Not Allowed With Header'.
A command was received which could only be a query and attempted to set some parameter, e.g. :COMMERROR 1.

6 'Query Not Allowed With Header'.
A command was received which could only be an action and tried to query some state or other.

7 'Parser Decode Error'.
Some part of the command did not meet the parser specification.

8 'Header Requires Sub-Header'.
The command is not complete, e.g. AUDIOANA

Example : `:COMmerror?`

Example response : `7`
Last error was a parser decode error

CONdition

:Freqstd?

Description : Returns the status of the currently-selected frequency standard; query only.

Response : <NR1> 0 OK
1 Out of lock
2 Absent

Example : `:CONdition:Freqstd?`

Example response : `1`
The selected frequency standard is out of lock

:COPy

Description : Performs a screen dump to a printer. Action only.

Parameters : <CPD> EPSON80
EPSON100
LASER100
LASER150
LASER300

Allowed suffices : Not applicable

Example : `:COPy LASER100`
Copies screen information to the HP Laserjet100 printer

:CWana

:MARkers

Description : Sets the status of the CW analyzer mode markers.

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:CWana:MARkers OFF`
Turn the CW analyzer markers off

:MARkers?

Description : Returns the current status of the CW analyzer mode markers.

Response : <CRD> OFF, ON

Example : `:CWana:MARkers?`

Example response : `CW:MAR OFF`

:CWana

:MAXhold

Description : Sets, releases, or resets, the Maximum Hold function on the CW analyzer

Parameters : <CPD> OFF
ON

Allowed suffices : Not applicable

Example : `:CWana:MAXhold ON`
Sets Maximum Hold function

:MAXhold?

Description : Returns the current setting of the Maximum Hold function

Response : <CRD> OFF
ON

Example : `:CWana:MAXhold?`

Example response : `CW:MAX ON`

:CWana

:MKR*n*

Description : Sets the frequency location of the CW analyzer marker *n*, where *n* = 1 or 2.

Parameters : <NRf>

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:CWana:MKR1 850MHZ`

:MKR*n*?

Description : Returns the current location of the CW analyzer marker *n*, where *n* = 1 or 2.

Response : <NR2> Units in Hz

Example : `:CWana:MKR1?`

Example response : `CW:MKR1 850000000`

:CWana

:Peakfind

Description : Causes the CW analyzer marker 1 to locate at the point of maximum signal level. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:CWana:Peakfind`
Places marker 1 to position of maximum displayed signal level

:CWana

:REFFreq

Description : Sets the CW analyzer reference frequency.

Parameters : <NRf>

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:CWana:REFFreq 850MHZ`

:REFFreq?

Description : Returns the current setting of the CW analyzer reference frequency.

Response : <NR2> Units in Hz

Example : `:CWana:REFFreq?`

Example response : `CW:REFF 850000000`

:CWana

:REFLevel

Description : Sets the reference level of the CW analyzer.

Parameters : <NRf>

Allowed suffices : DB

Example : `:CWana:REFLevel -20DB`
Set the CW analyzer reference level to –20 dB.

:REFLevel?

Description : Returns the current setting of the CW analyzer reference level in dB.

Response : <NR2> Units in dB

Example : `:CWana:REFLevel?`

Example response : `CW:REFL -20DB`

:CWana

:SIze

Description : Sets the view of the CW analyzer display to normal or expanded.

Parameters : <CPD> NORMAL, EXPAND

Allowed suffices : Not applicable

Example : `:CWana:SIze NORMAL`
Set the CW analyzer display to normal view

:SIze?

Description : Returns the current setting of the CW analyzer display view.

Response : <CRD> NORMAL, EXPAND

Example : `:CWana:SIze?`

Example response : `CW:SI NORMAL`

:CWana

:SPan

Description : Sets the frequency span/division of the CW analyzer mode.

Parameters : <CPD> AS_5HZ
AS_10HZ
AS_25HZ
AS_50HZ
AS_100HZ
AS_250HZ
AS_500HZ
AS_1KHZ
AS_2_5KHZ
AS_5KHZ

Allowed suffices : Not applicable

Example : `:CWana:SPan AS_5KHZ`
Set the CW analyzer span to 5 kHz/division

:SPan?

Description : Returns the current setting of the CW analyzer frequency span/division.

Response : <CRD> AS_5HZ
AS_10HZ
AS_25HZ
AS_50HZ
AS_100HZ
AS_250HZ
AS_500HZ
AS_1KHZ
AS_2_5KHZ
AS_5KHZ

Example : `:CWana:SPan?`

Example response : `CW:SP AS_5KHZ`

:CWana

:Trig

Description : Controls the CW analyzer trigger mode.

Parameters : <CPD> SINGLE, REPEAT

Allowed suffices : Not applicable

Example : `:CWANA:Trig SINGLE`
Set the CW analyzer trigger mode to single sweep

:Trig?

Description : Returns the current setting of the CW analyzer trigger mode.

Response : <CRD> SINGLE, REPEAT

Example : `:CWANA:Trig?`

Example response : `CW:T Single`

:CWana

:Vrange

Description : Sets the vertical range/division of the CW analyzer.

Parameters : <NRf>

Allowed suffices : DB

Example : :CWana:Vrange 2DB
Sets the CW analyzer vertical scale to 2 dB/division

:Vrange?

Description : Returns the current vertical range/division of the CW analyzer.

Response : <NR2> Units in dB

Example : :CWana:Vrange?

Example response : CW:V 2.000DB

:DAFmode

Description : Sets the Digital AF TEST screen when in Digital AF TEST mode.

Parameters : <CPD> MENU
NORMAL
MULTIMETER

Allowed suffices : Not applicable

Example : :DAFmode NORMAL
Set the instrument into the normal audio screen when in digital audio test mode.

:DAFmode?

Description : Returns the current selection of the Digital AF TEST screen.

Response : <CRD> MENU
NORMAL
MULTIMETER

Example : :DAFmode?

Example response : DAF NORMAL_

:DATe

Description : Sets the date for the real-time clock.

Parameters : <NR1>,<NR1>,<NR1> Day, month, year in format *dd, mm, yy*

Allowed suffices : Not applicable

Example : `:DATe 27,10,95`
Sets the real-time clock date to 27 October 1995

:DATe?

Description : Returns the current setting of the real-time clock date.

Response : <NR1>, <NR1>, <NR1>

Example : `:DATe?`

Example response : `DAT 27,9,95`

:DDxmode

Description : Sets the type of Digital DUPLEX TEST screen, when already in a Digital DUPLEX TEST mode.

Parameters : <CPD> NORMAL
TX_DISPLAY
RX_DISPLAY

Allowed suffices : Not applicable

Example : `:DDxmode TX_DISPLAY`
Set the instrument to the expanded Tx screen when in Digital DUPLEX TEST mode

:DDxmode?

Description : Returns the current setting of Digital DUPLEX TEST screen.

Response : <CRD> NORMAL
TX_DISPLAY
RX_DISPLAY

Example : `:DDxmode?`

Example response : `DD TX_DISPLAY`

:DEModtype

Description : Sets the RF signal demodulation type.

Parameters : <CPD> AM
FM
PM

Allowed suffices : Not applicable

Example : `:DEModtype AM`

Select AM as the demodulation method

:DEModtype?

Description : Returns the current demodulation type.

Response : <CRD> AM
FM
PM

Example : `:DEModtype?`

Example response : `DEM AM`

:DEVerror?

Description : Returns the last device error generated by the instrument.

Response : <NR1>

0 'No Error'

1 'Instrument Range or Setting Error'

Some command was received that the instrument could not process due to out of range data or the instrument not being in the correct mode to action the requested command.

2 'Request For Measurement Pending'
Indicates that another measurement has been requested, before the previous one has been completed

3 'Calibration already running'
Indicates that calibration is already in progress when a calibration was requested.

4 'Systems Test already running'
Indicates that a Systems Auto test was already in progress when another Auto test was requested.

5 'Digital measurements not valid'
A digital measurement has been requested and the result is not valid, e.g. no bursts.

6 'Measurement invalid in this mode'
A measurement has been requested that cannot be performed in the current test mode. For example, a modulation measurement in an audio test mode.

7 'Frequency Standard Error'
There is a problem with the currently selected frequency standard. It is either absent or out of lock.

Example : `:DEVerror?`

Example response : `1`
Instrument range or setting error

:DIsplay

Description : Sets the display presentation method.

Parameters : <CPD> BARCHARTS
FFT (Audio Ana)
VECTORDIAG
PHASEANA
POWERANA
SCOPE
SPECANA

Note: VECTORDIAG *is only available if a TETRA system has been selected,* PHASEANA *is only available if GSM, DCS or PCS system has been selected and* POWERANA *is only available if GSM, DCS, PCS or a TETRA system has been selected.*

Allowed suffices : Not applicable

Example : `:DIsplay BARCHARTS`
Sets the inset display type to bar charts

:DIsplay?

Description : Returns the current setting of the display presentation.

Response : <CRD> BARCHARTS
FFT
VECTORDIAG
PHASEANA
POWERANA
SCOPE
SPECANA
NO_DISPLAY

Example : `:DIsplay?`

Example response : `DI BARCHARTS`

:DUplexmode

Description : Sets the type of DUPLEX TEST screen, when already in a DUPLEX TEST mode or sets the type of DIGITAL DUPLEX TEST screen when already in a DIGITAL DUPLEX TEST mode.

Parameters : <CPD> NORMAL
TX_DISPLAY
RX_DISPLAY
ACC_DPOWER
DX_TONES_SEQ_GEN
DX_TONES_SEQ_DEC
DX_TONES_CTCSS
DX_TONES_DTMF
DX_TONES_DTMF_GEN
DX_TONES_DTMF_DEC
DX_TONES_DCS

Note: ACC_DPOWER *and the* DX TONES *screens are not available in* DIGITAL DUPLEX TEST mode.

Allowed suffices : Not applicable

Example : `:DUplexmode TX_DISPLAY`
Set the instrument to the DUPLEX TEST Tx *screen when in* DUPLEX TEST *mode or to the* DIGITAL DUPLEX TEST Tx *screen in* DIGITAL DUPLEX TEST *mode.*

:DUplexmode?

Description : Returns the current setting of DUPLEX TEST screen.

Response : <CRD> NORMAL
TX_DISPLAY
RX_DISPLAY
ACC_DPOWER
DX_TONES_SEQ_GEN
DX_TONES_SEQ_DEC
DX_TONES_CTCSS
DX_TONES_DTMF
DX_TONES_DTMF_GEN
DX_TONES_DTMF_DEC
DX_TONES_DCS

Note: ACC_DPOWER *and the* DX TONES *screens are not available in* DIGITAL DUPLEX TEST mode.

Example : `:DUplexmode?`

Example response : `DU TX_DISPLAY`

:EXecerror?

Description : Returns the type of the last error generated by the Execution control routine.

Response : <NR1>

0 'No Error'.

1 'Data Type Not Allowed'.
A command which takes <CPD> or <NRf> in a one of few form has been sent with a number larger than that recognised as the highest possible, e.g. :SCOPE:TRIG 5

2 'Excess Data'.
More data was received with the command than was expected, e.g. :AFGEN1:FR 10.000,15.000

3 'Insufficient Data'.
The command had fewer data fields than expected.

4 'Some Data Required'.
No data came with the command when some was definitely required, e.g. :AFGEN1:FREQ.

5 'No Text Match Found'
<CPD> was received which did not tally with the allowed character data strings for that data field e.g.: SCOPE:AFRANGE GARBLE

6 'Text Not Unique'.
The abbreviated <CPD> received was too short to be uniquely recognised e.g. : SCOPE:AFRANGE SC_2

7 'Suffix Not Found'.
The suffix received with a particular data field was not one allowed for that command, e.g. :AFGEN1:FREQ 10.000DBM.

8 'Suffix Not Allowed'.
A suffix was sent with a numeric data field when one was not allowed
e.g. :SCOPE:AFRANGE 5KHZ.

9 'Query Error'.
Query could not be executed, e.g.
:MEMCARD:FORMAT? when no card is inserted.

Example : `:EXecerror?`

Response : 2
More data has been sent than was expected

:Hwreset

Description : Hardware reset; has the same effect as switching the test set OFF and ON again. Action only. For more information, see the *Reset levels* section in Chapter 4.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:Hwreset`
Produces a switch-on type reset.

:MEASAverages

:Amps

Description : Sets the number of averages for Ampere measurements.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:Amps 5`
Set the Ampere measurement averages to 5

:Amps?

Description : Returns the current setting of the number of measurement averages for Amperes.

Response : <NR1>

Example : `:MEASAverages:Amps?`

Example response : `MEASA:A 5`

:MEASAverages

:BER1

Description : Sets the number of averages (samples) taken for digital bit error rate class 1 measurements.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:BER1 10`
Set the digital bit error rate class 1 measurement averages to 10

:BER1?

Description : Returns the current setting of the number of measurement averages for bit error rate class 1.

Response : <NR1>

Example : `:MEASAverages:BER1?`

Example response : `MEASA:BER1 10`

:MEASAverages

:BER2

Description : Sets the number of averages (samples) taken for digital bit error rate class 2 measurements.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:BER2 10`
Set the digital bit error rate class 2 measurement averages to 10

:BER2?

Description : Returns the current setting of the number of measurement averages for bit error rate class 2.

Response : <NR1>

Example : `:MEASAverages:BER2?`

Example response : `MEASA:BER2 10`

:MEASAverages

:Digtxtype

Description : Sets the digital Tx measurement type for the following measurements:-
frequency (see :MEASAverages:FReqerr, page 5-107),
peak phase error (see :MEASAverages:PHASEPeak, page 5-109),
RMS phase error (see :MEASAverages:PHASERms, page 5-109) ,
peak vector error, (see :MEASAverages:VECTPeak, page 5-111),
RMS vector error (see :MEASAverages:VECTRMs, page 5-112),
residual carrier (see :MEASAverages:VECTRMs, page 5-111)
and
power (see :MEASAverages:POwer, page 5-108)

Parameters : <CPD> WORSTCASE, AVERAGE

Allowed suffices : Not applicable

Example : `:MEASAverages:Digtxtype AVERAGE`
Sets the digital Tx measurement type to average

:Digtxtype?

Description : Returns the current setting of the digital Tx measurement type.

Response : <CRD> WORSTCASE, AVERAGE

Example : `:MEASAverages:Digtxtype?`

Example response : `MEASA:D AVERAGE`

:MEASAverages

:FEr

Description : Sets the number of averages (samples) taken for frame erasure rate measurements.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:FEr 10`
Set the frame erasurerate measurement averages to 10

:FEr?

Description : Returns the current setting of the number of measurement averages for frame erasure rate.

Response : <NR1>

Example : `:MEASAverages:FEr?`

Example response : `MEASA:FE 10`

:MEASAverages

:FReqerr

Description : Sets the number of averages (samples) taken for digital frequency error measurements. The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, it can be set via *[digital systems]* in HELP SET-UP.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:FReqerr 10`
Set the digital frequency error measurement averages to 10

:FReqerr?

Description : Returns the current setting of the number of measurement averages for digital frequency error.

Response : <NR1>

Example : `:MEASAverages:FReqerr?`

Example response : `MEASA:FR 10`

:MEASAverages

:Ohms

Description : Sets the number of averages for Ohms measurements.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:Ohms 5`
Set the Ohms measurement averages to 5

:Ohms?

Description : Returns the current setting of the number of measurement averages for Ohms.

Response : <NR1>

Example : `:MEASAverages:Ohms?`

Example response : `MEASA:O 5`

:MEASAverages

Applicable to TETRA MOBILE, TETRA BASE and TETRA DIRECT

:POwer

Description : Sets the number of averages (samples) taken for digital power measurements. The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, it can be set via *[digital systems]* in HELP SET-UP.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:POwer 10`
Set the digital power measurement averages (samples) to 10

:POwer?

Description : Returns the current setting of the number of measurement averages (samples) for digital power.

Response : <NR1>

Example : `:MEASAverages:POwer?`

Example response : `MEASA:PO 10`

:MEASAverages

:PHASEPeak

Description : Sets the number of averages (samples) taken for digital peak phase measurements. The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, it can be set via *[digital systems]* in HELP SET-UP.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:PHASEPeak 10`
Set the peak phase measurement averages to 10

:PHASEPeak?

Description : Returns the current setting of the number of measurement averages for peak phase.

Response : <NR1>

Example : `:MEASAverages:PHASEPeak?`

Example response : `MEASA:PHASEP 10`

:MEASAverages

:PHASERms

Description : Sets the number of averages (samples) taken for digital RMS phase measurements. The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, it can be set via *[digital systems]* in HELP SET-UP.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:PHASERms 10`
Set the peak phase measurement averages to 10

:PHASERms?

Description : Returns the current setting of the number of measurement averages for RMS phase.

Response : <NR1>

Example : `:MEASAverages:PHASERms?`

Example response : `MEASA:PHASER 10`

:MEASAverages

:RBER1b

Description : Sets the number of averages (samples) taken for digital residual bit error rate class 1 measurements.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:RBER1b 10`
Set the digital residual bit error rate class 1 measurement averages to 10

:RBER1b?

Description : Returns the current setting of the number of measurement averages for residual bit error rate class 1.

Response : <NR1>

Example : `:MEASAverages:RBER1b?`

Example response : `MEASA:RBER1 10`

:MEASAverages

:RBER2

Description : Sets the number of averages (samples) taken for digital residual bit error rate class 2 measurements.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:RBER2 10`
Set the digital residual bit error rate class 2 measurement averages to 10

:RBER2?

Description : Returns the current setting of the number of measurement averages for residual bit error rate class 2.

Response : <NR1>

Example : `:MEASAverages:RBER2?`

Example response : `MEASA:RBER2 10`

:MEASAverages

:VECTPeak

Description : Sets the number of averages (samples) taken for peak vector error measurements. The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, it can be set via *[digital systems]* in HELP SET-UP.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:VECTPeak 10`
Set the peak vector error measurement averages (samples) to 10

:VECTPeak?

Description : Returns the current setting of the number of measurement averages (samples) for peak vector error.

Response : <NR1>

Example : `:MEASAverages:VECTPeak?`

Example response : `MEASA:VECTP 10`

:MEASAverages

:VECTREs

Description : Sets the number of averages (samples) taken for residual carrier measurements. The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, it can be set via *[digital systems]* in HELP SET-UP.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:VECTREs 10`
Set the residual carrier measurement averages (samples) to 10

:VECTREs?

Description : Returns the current setting of the number of measurement averages (samples) for residual carrier.

Response : <NR1>

Example : `:MEASAverages:VECTREs?`

Example response : `MEASA:VECTRE 10`

:MEASAverages

:VECTRMs

Description : Sets the number of averages (samples) taken for RMS vector error measurements. The Tx measurement type (worst case or average) can be set by :MEASAverages:Digtxtype (page 5-106). Alternatively, it can be set via *[digital systems]* in HELP SET-UP.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:MEASAverages:VECTRMs 10`
Set the RMS vector error measurement averages (samples) to 10

:VECTRMs?

Description : Returns the current setting of the number of measurement averages (samples) for RMS vector error.

Response : <NR1>

Example : `:MEASAverages:VECTRMs?`

Example response : `MEASA:VECTRM 10`

:MEASCycle

Description : The measurement cycle mode is the way in which measurements are returned over the remote control. When the measurement cycle is ON, the remote control assumes that they have been stabilised, and so does not wait for the next measurement of the type requested to finish before returning the value. Instead, it simply returns the value that was last read.

This means that the measurements are returned more quickly. However, it should only be switched ON when it is *known* that the measurements have stabilised, otherwise inaccurate results might be returned.

Switching it OFF means that a measurement request causes a new measurement to be taken before it is returned.

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:MEASCycle ON`
Set the measurement cycle to ON

:MEASCycle?

Description : Returns the current status of the measure cycle

Response : <CRD> OFF, ON

Example : `:MEASCycle?`

Example response : `MEASC ON`

:MODAna

:AMreflevel

Description : Sets the amplitude modulation reference level of the modulation analyzer.

Parameters : <NRf>

Allowed suffices : PC

Example : :MODAna:AMreflevel 80PC

:AMreflevel?

Description : Returns the current setting of the amplitude modulation reference level of the modulation analyzer.

Response : <NR2> Units in %

Example : :MODA:AMrelevel?

Example response : MODA:AM 80PC

:MODAna

:Fmreflevel

Description : Sets the frequency modulation reference level of the modulation analyzer.

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : :MODAna:Fmreflevel 15KHZ

:Fmreflevel?

Description : Returns the current setting of the frequency modulation reference level of the modulation analyzer.

Response : <NR2> Units in Hz

Example : :MODA:Fmreflevel?

Example response : MODA:F 15000HZ

:MODAna

:MARkers

Description : Sets the status of the modulation analyzer markers.

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:MODAna:MARkers OFF`
Turn the modulation analyzer markers OFF

:MARkers?

Description : Returns the current status of the modulation analyzer markers.

Response : <CRD> OFF, ON

Example : `:MODA:MARkers?`

Example response : `MODA:MAR OFF`

:MODAna

:MAXhold

Description : Sets, releases, or resets, the Maximum Hold function on the modulation analyzer

Parameters : <CPD> OFF
ON

Allowed suffices : Not applicable

Example : `:MODAna:MAXhold ON`
Sets Maximum Hold function

:MAXhold?

Description : Returns the current setting of the Maximum Hold function

Response : <CRD> OFF
ON

Example : `:MODAna:MAXhold?`

Example response : `MODA:MAX ON`

:MODAna

:MKR*n*

Description : Sets the frequency location of modulation analyzer marker *n*, where *n* = 1 or 2.

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : :MODAna:MKR1 25.5KHZ
Set audio analyzer marker 1 to 25.5 kHz

:MKR*n*?

Description : Returns the current location of modulation analyzer marker *n*, where *n* = 1 or 2.

Response : <NR2> Units in Hz

Example : :MODAna:MKR1?

Example response : MODA:MKR1 25500HZ.

:MODAna

:PEakfind

Description : Causes the or modulation analyzer marker 1 to locate at the point of maximum signal level. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : :MODAna:PEakfind
Locate audio analyzer marker 1 at the frequency where the peak modulation level is currently found.

:MODAna

:PMreflevel

Description : Sets the phase modulation reference level of the modulation analyzer.

Parameters : <NRf>

Allowed suffices : RAD, R

Example : :MODAna:PMreflevel 50RAD

:PMreflevel?

Description : Returns the current setting of the phase modulation reference level of the modulation analyzer.

Response : <NR2> Units in radians

Example : :MODAna:PMreflevel?

Example response : MODA:PM 50RAD

:MODAna

:Reffreq

Description : Sets the modulation analyzer reference frequency.

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : :MODAna:Reffreq 25.5KHZ
Set the modulation analyzer reference frequency to 25.5 kHz.

:Reffreq?

Description : Returns the current setting of the modulation analyzer reference frequency.

Response : <NR2> Units in Hz

Example : :MODAna:Reffreq?

Example response : MODA:REF 25500HZ

:MODAna

:SIze

Description : Sets the view of the modulation analyzer to normal or expanded.

Parameters : <CPD> NORMAL, EXPAND

Allowed suffices : Not applicable

Example : :MODAna:SIze NORMAL
Set the modulation analyzer display to normal view.

:SIze?

Description : Returns the current setting of the view of the audio analyzer (or modulation analyzer) mode.

Response : <CRD> NORMAL, EXPAND

Example : :MODAna:SIze?

Example response : MODA:SI NORMAL

:MODAna

:SPan

Description : Sets the frequency span/division of the modulation analyzer.

Parameters : <CPD> AS_5HZ
AS_10HZ
AS_25HZ
AS_50HZ
AS_100HZ
AS_250HZ
AS_500HZ
AS_1KHZ
AS_2_5KHZ
AS_5KHZ

Allowed suffices : Not applicable

Example : `:MODAna:SPan AS_5HZ`
Set the modulation analyzer span/division to 5 kHz

:SPan?

Description : Returns the current setting of the modulation analyzer frequency span/division.

Response : <CRD> AS_5HZ
AS_10HZ
AS_25HZ
AS_50HZ
AS_100HZ
AS_250HZ
AS_500HZ
AS_1KHZ
AS_2_5KHZ
AS_5KHZ

Example : `:MODAna:SPan?`

Example response : `MODA:SP AS_5HZ`

:MODAna

:Trig

Description : Controls the modulation analyzer trigger mode.

Parameters : <CPD> SINGLE, REPEAT

Allowed suffices : Not applicable

Example : `:MODAna:Trig SINGLE`
Set the modulation analyzer trigger mode to single sweep.

:Trig?

Description : Returns the current setting of the modulation analyzer trigger mode.

Response : <CRD> SINGLE, REPEAT

Example : `:MODAna:Trig?`

Example response : `MODA:T SINGLE`

:MODAna

:Vrange

Description : Sets the vertical span of the modulation analyzer.

Parameters : <NRf>

Allowed suffices : DB

Example : `:MODAna:Vrange 2DB`
Sets the vertical scaling to 2 dB per division

:Vrange?

Description : Returns current setting of vertical scaling of modulation analyzer.

Response : <NR2> Units in dB

Example : `:MODAna:Vrange?`

Example response : `MODA:V 2DB`

:MODGEN*n*

:Amdepth

Description : Sets the amplitude modulation depth of modulation generator *n* where *n* = 1 to 4

Parameters : <NRf>

Allowed suffices : PC

Example : `:MODGEN1:Amdepth 30PC`
Set the amplitude modulation depth of modulation generator 1 to 30%

:Amdepth?

Description : Returns the current setting of the amplitude modulation depth of modulation generator *n*, where *n* = 1 to 4, in %.

Response : <NR2> Units in %

Example : `:MODGEN1:Amdepth?`

Example response : `MODGEN1:A 30PC`

:MODGEN*n*

:Display

Description : This command displays the selected modulation generator parameters if an appropriate screen is in use. *n* = 1 to 4. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:MODGEN3:Display`

:MODGEN*n*

:FMdevn

Description : Sets the frequency deviation of modulation generator *n* where *n* = 1 to 4

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : :MODGEN1:FMdevn 3.5KHZ
Set the frequency modulation deviation of modulation generator 1 to 3.5 kHz

:FMdevn?

Description : Returns the current setting of modulation generator *n* where *n* = 1 to 4

Response : <NR2> Frequency in Hz

Example : :MODGEN1:FMdevn?

Example response : MODGEN1:FM 3500HZ

:MODGEN*n*

:FReq

Description : Sets the frequency of modulation generator *n* where *n* = 1 to 4

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : :MODGEN1:FReq 2.5KHZ
Set modulation generator 1 to 2.5 kHz

:FReq?

Description : Returns the current setting of modulation generator *n* where *n* = 1 to 4

Response : <NR2> Frequency in Hz

Example : :MODGEN1:FReq?

Example response : MODGEN1:FR 2500HZ

:MODGEN*n*

:Pmdevn

Description : Sets the amplitude of the phase deviation of modulation generator *n* where *n* = 1 to 4

Parameters : <NRf>

Allowed suffices : RAD, R

Example : :MODGEN1:Pmdevn 5RAD
Set the phase modulation of modulation generator 1 to 50 radians

:Pmdevn?

Description : Returns the current setting of the phase deviation of modulation generator *n* where *n* = 1 to 4.

Response : <NR2> Units in radians

Example : :MODGEN1:Pmdevn?

Example response : MODGEN1:P 5RAD

:MODGEN*n*

:Status

Description : Sets the status of modulation generator *n* where *n* = 1 to 4

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : :MODGEN1:Status ON
Set the status of modulation generator 1 to ON

:Status?

Description : Returns the current status of modulation generator *n* where *n* = 1 to 4

Response : <CRD> OFF, ON

Example : :MODGEN1:Status?

Example response : MODGEN1:S ON

:MODGENX

:Amdepth

Description : Sets external modulation AM depth

Parameters : <NRf>

Allowed suffices : PC

Example : :MODGENX:Amdepth 30PC
Set the amplitude modulation level of external modulation to 30

:Amdepth?

Description : Returns the current setting of external modulation amplitude modulation depth

Response : <NR2> Units in %

Example : :MODGENX:Amdepth?

Example response : MODGENX:A 30PC

:MODGENX

:Display

Description : This command displays the external modulation generator parameters if an appropriate screen is in use. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : :MODGENX:Display

:MODGENX

:Fmdevn

Description : Sets external modulation FM deviation level

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : :MODGENX:Fmdevn 3.5KHZ
Set the frequency modulation level of external modulation to 3.5 kHz

:Fmdevn?

Description : Returns the current setting of external frequency modulation deviation.

Response : <NR2> Units in Hz

Example : :MODGENX:Fmdevn?

Example response : MODGENX:F 3500HZ

:MODGENX

:Pmdevn

Description : Sets external phase modulation deviation
Parameters : <NRf>
Allowed suffices : RAD
Example : :MODGENX:Pmdevn 2RAD
Set the phase modulation level of external modulation to 2 radians

:Pmdevn?

Description : Returns the current setting of external phase modulation deviation.
Response : <NR2> Units in radians
Example : :MODGEN1:Pmdevn?
Example response : MODGEN1:P 2RAD

:MODGENX

:Status

Description : Sets external modulation status
Parameters : <CPD> OFF, ON
Allowed suffices : Not applicable
Example : :MODGENX:Status ON
Set the status of external modulation to ON

:Status?

Description : Returns the current status of external modulation
Response : <CRD> OFF, ON
Example : :MODGENX:Status?
Example response : MODGENX:S ON

:MODInput

Description : Controls the coupling of the EXT MOD INPUT socket

Parameters : <CPD> AC, DC

Allowed suffices : Not applicable

Example : `:MODInput:AC`
Set the modulation input coupling to AC

:MODInput?

Description : Returns the coupling of the EXT MOD INPUT socket

Response : <CRD> AC, DC

Example : `:MODInput?`

Example response : `MODI AC`

:MODType

Description : Sets the current modulation type

Parameters : <CPD> AM
FM
PM

Allowed suffices : Not applicable

Example : `:MODType AM`

:MODType?

Description : Returns the current modulation type

Response : <CRD> AM
FM
PM

Example : `:MODType?`

Example response : `MODT AM`

:MODUlation

Description : Controls the status of the overall modulation (analogue or digital depending on the current mode). An OFF command overrules an ON command for an individual modulation generator (e.g. :MODGEN1:STATUS ON)

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:MODUlation ON`
Switches the modulation generator function ON

:MODUlation?

Description : Returns the current setting of the modulation overall state

Response : <CRD> OFF, ON

Example : `:MODUlation?`

Example response : `MODU ON`

:MUltimeter

:ACAmps

Description : Sets the multimeter mode AC amps range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
ACA_1A
ACA_10A

Allowed suffices : Not applicable

Example : `:MUltimeter:ACAmps ACA_1A`
Set the multimeter AC amps range to 1 AMP

:ACAmps?

Description : Returns the current setting of the multimeter mode AC amps range

Response : <CPD> ACA_1A
ACA_10A

Example : `:MUltimeter:ACAmps?`

Example response : `MU:ACA ACA_1A`

:MUltimeter

:ACVolts

Description : Sets the multimeter mode AC volts range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
ACV_300MV
ACV_3V
ACV_30V
ACV_300V

Allowed suffices : Not applicable

Example : `:MUltimeter:ACVolts ACV_300MV`
Set the multimeter AC range to 300 mV

:ACVolts?

Description : Returns the current setting of the multimeter mode AC volts range

Response : <CRD> ACV_300MV
ACV_3V
ACV_30V
ACV_300V

Example : `:MUltimeter:ACVolts?`

Example response : `MU:ACV ACV_300MV`

:MUltimeter

:DCAmps

Description : Sets the multimeter mode DC amps range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
DCA_1A
DCA_10A

Allowed suffices : Not applicable

Example : `:MUltimeter:DCAmps DCA_1A`
Set the multimeter DC amps range to 1 AMP

:DCAmps?

Description : Returns the current setting of the multimeter mode DC amps range

Response : <CRD> DCA_1A
DCA_10A

Example : `:MUltimeter:DCAmps?`

Example response : `MU:DCA DCA_1A`

:MUltimeter

:DCVolts

Description : Sets the multimeter mode DC volts range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
DCV_300MV
DCV_3V
DCV_30V
DCV_300V
DCV_750V

Allowed suffices : Not applicable

Example : `:MUltimeter:DCVolts DCV_300MV`
Set the multimeter DC range to 300 mV

:DCVolts?

Description : Returns the current setting of the multimeter mode DC volts range

Response : <CRD> DCV_300MV
DCV_3V
DCV_30V
DCV_300V
DCV_750V

Example : `:MUltimeter:DCVolts?`

Example response : `MU:DCV DCV_300MV`

:MUltimeter

:Maxhold

Description : Sets, releases, or resets, the Maximum Hold function on the multimeter

Parameters : <CPD> OFF
ON
HOLD
RESET

Allowed suffices : Not applicable

Example : `:MUltimeter:Maxhold HOLD`
Sets Maximum Hold function on resistance ranges

:Maxhold?

Description : Returns the current setting of the Maximum Hold function on the multimeter

Response : <CRD> OFF
ON
HOLD
RESET

Example : `:MUltimeter:Maxhold?`

Example response : `MU:M HOLD`

:MUltimeter

:Ohms

Description : Sets the multimeter mode resistance measuring range to auto-ranging or to a defined range.

Parameters : <CPD> AUTO
OHM_100
OHM_1K0
OHM_10K
OHM_100K
OHM_1M0

Allowed suffices : Not applicable

Example : `:MUltimeter:Ohms OHM_100`
Set the multimeter ohms range to 100 Ω

:Ohms?

Description : Returns the current setting of the multimeter mode Ohms range

Response : <CRD> OHM_100
OHM_1K0
OHM_10K
OHM_100K
OHM_1M0

Example : `:MUltimeter:Ohms?`

Example response : `MU:O OHM_100`

:MUltimeter

:Type

Description : Sets the multimeter mode measurement type

Parameters : <CPD> DCVOLTS
ACVOLTS
DCAMPS
ACAMPS
OHMS
CONTINUITY

Allowed suffices : Not applicable

Example : `:MUltimeter:Type DCVOLTS`
Set the multimeter mode to measure DC volts

:Type?

Description : Returns the current multimeter mode measurement type

Response : <CRD> DCVOLTS
ACVOLTS
DCAMPS
ACAMPS
OHMS
CONTINUITY

Example : `:MUltimeter:Type?`

Example response : `MU:T DCVOLTS`

:PHaseana

:MArkers

Description : Set the state of the phase analyzer mode markers

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PHaseana:MArkers ON`
Turn the phaseana markers on.

:MArkers?

Description : Returns the current setting of the state of the phase analyzer markers

Response : <CRD> OFF, ON

Example : `:PHaseana:MArkers?`

Example response : `PH:MA ON`

:PHaseana

:MKR*n*

Description : .Sets the position of the phase analyzer marker *n*, where *n* = 1 or 2

Parameters : <NRf>

Allowed suffices : BITS

Example : `:PHaseana:MKR1 100BITS`
Set the phase anal;yser marker 1 to position 100 Bits

:MKR*n*?

Description : Returns the current setting of the position of phase analyzer marker *n*, where *n* = 1 or 2

Response : <NR2> units in BITS

Example : `:PHaseana:MKR1?`

Example response : `PH:MKR1 100BITS`

:PHaseana

:SCale

Description : Set the vertical scale for the phase analyzer.

Parameters : <CPD> PHANA_2_5DEG
PHANA_5DEG
PHANA_10DEG

Allowed suffices : Not applicable

Example : `:PHaseana:SCale PHANA_10DEG`
Set the phase analyzer vertical scale to 10° per division

:SCale?

Description : Returns the current setting of the phase analyzer scale

Response : <CRD> PHANA_2_5DEG
PHANA_5DEG
PHANA_10DEG

Example : `:PHaseana:SCale?`

Example response : `PH:SC PHANA_10DEG`

:PHaseana

:SIze

Description : Set the size of the phase analyzer to normal or expanded.

Parameters : <CPD> NORMAL
EXPAND

Allowed suffices : Not applicable

Example : `:PHaseana:SIze EXPAND`
Set the phase analyzer to expanded view

:SIze?

Description : Returns the current setting of the phase analyzer size.

Response : <CRD> NORMAL
EXPAND

Example : `:PHaseana:SIze?`

Example response : `PH:SI EXPAND`

:PHaseana

:Trig

Description : Set the phase analyzer trigger mode

Parameters : <CPD> SINGLE, REPEAT

Allowed suffices : Not applicable

Example : `:PHaseana:Trig REPEAT`
Set the phase analyzer trigger mode to repeat

:Trig?

Description : Returns the current setting of the phase analyzer trigger mode

Response : <CRD> SINGLE, REPEAT

Example : `:PHaseana:Trig?`

Example response : `PH:T REPEAT`

:PWrana

:MArkers

Description : .Set the state of the power analyzer mode markers

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PWrana:MArkers ON`
Turn the powerana markers on.

:MArkers?

Description : Returns the current setting of the state of the power analyzer markers

Response : <CRD> OFF, ON

Example : `:PWrana:MArkers?`

Example response : `PW:MA ON`

:PWrana

:MKR*n*

Description : .Sets the position of the power analyzer marker *n* where *n* = 1 or 2

Parameters : <NRf>

Allowed suffices : `BITS, BIT`

Example : `:PWrana:MKR1 100BITS`
Set the power anal;yser marker 1 to position 100 Bits

:MKR*n*?

Description : Returns the current setting of the position of power analyzer marker *n* where *n* = 1 or 2

Response : <NR2> units in BITS

Example : `:PWrana:MKR1?`

Example response : `PW:MKR1 100BITS`

:PWrana

Applicable to TETRA MOBILE, TETRA BASE and TETRA DIRECT systems

:MKRSYM*n*

Description : .Sets the position of the power analyser marker *n* where *n* = 1 or 2

Parameters : <NRf>

Allowed suffices : `SYMBOLS, SYMS, SYM`

Example : `:PWrana:MKRSYM1 100SYMBOLS`
Set the power anal;yser marker 1 to position 100 Symbols

:MKRSYM*n*?

Description : Returns the current setting of the position of power analyser marker *n* where *n* = 1 or 2

Response : <NR2> units in SYMBOLS

Example : `:PWrana:MKRSYM1?`

Example response : `PW:MKRSYM1 100SYMBOLS`

:PWrana

Applicable to TETRA MOBILE, TETRA BASE and TETRA DIRECT systems

:SCale

Description : Set the vertical scale for the power analyser.

Parameters : <CPD> PWRANA_3DB
PWRANA_10DB

Allowed suffices : Not applicable

Example : `:PWrana:SCale PWRANA_10DB`
Set the power analyser vertical scale to 10dB per division

:SCale?

Description : Returns the current setting of the power analyser scale

Response : <CPD> PWRANA_3DB
PWRANA_10DB

Example : `:PWrana:SCale?`

Example response : `PW:SC PWRANA_10dB`

:PWrana

:SIze

Description : Set the size of the power analyzer to normal or expanded.

Parameters : <CPD> NORMAL
EXPAND

Allowed suffices : Not applicable

Example : `:PWrana:SIze EXPAND`
Set the power analyzer to expanded view

:SIze?

Description : Returns the current setting of the power analyzer size.

Response : <CRD> NORMAL
EXPAND

Example : `:PWrana:SIze?`

Example response : `PW:S EXPAND`

:PWrana

:TRig

Description : Set the power analyzer trigger mode

Parameters : <CPD> SINGLE, REPEAT

Allowed suffices : Not applicable

Example : `:PWrana:TRig REPEAT`
Set the power analyzer trigger mode to repeat

:TRig?

Description : Returns the current setting of the power analyzer trigger mode

Response : <CRD> NORMAL
EXPAND

Example : `:PWrana:TRig?`

Example response : `PW:TR REPEAT`

:PWrana

:TYpe

Description : .Set the type of display for the power analyser

Parameters : <CPD> FULLBURST, USEFUL_PART or RAMPS
(for GSM, DCS 1800 and PCS1900 systems)

or FULLBURST or RAMPS (for TETRA systems)

Allowed suffices : Not applicable

Example : `:PWrana:TYpe RAMPS`
Set the power analyser display to ramps

:TYpe?

Description : Returns the current setting of the power analyser type

Response : <CPD> FULLBURST, USEFUL_PART or RAMPS
(for GSM, DCS 1800 and PCS1900 systems)

or FULLBURST or RAMPS (for TETRA systems)

Example : `:PWrana:TYpe?`

Example response : `PW:TY RAMPS`

:Qerror?

Description : Returns the last query error generated by the Message Exchange Protocol Enforcer

Response : <NR1>

0 'No Error'.

1 'Interrupted'
If the instrument receives a data type following a query message, but has not sent the response message terminator, then it is 'interrupted'

2 'Unterminated'
If the controller attempts to read a message from the instrument before sending a complete query, then it is 'uninterrupted'

3 'Deadlocked'
When the output queue is full, no messages can be put in the output buffer, therefore no messages can be passed from the input buffer. If this is also full and the controller is trying to send a message byte, the instrument is 'deadlocked'

Example : `:Qerror?`

Example response : `3`
Instrument is deadlocked

:RECAll

Description : Recalls an instrument configuration from an internal memory store or from a memory location on an external memory card. Action only.

Parameters : <NR1> n = 0 to 34

Allowed suffices : Not applicable

Example : `:RECAll 21`
Restores instrument to state defined in store 21 on memory card

:RECEiver

:Autotune

Description : Controls the autotune function of the test set receiver

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:RECEiver:Autotune OFF`
Switches the autotune feature off

:Autotune?

Description : Returns the current status of the receiver autotune function

Response : <CAD> OFF, ON

Example : `:RECEiver:Autotune?`

Example response : `RECE:A OFF`

:RECEiver

:Bursttype

Description : Sets the type of TETRA burst to be measured

Parameters : <CPD> CONTROL, NORMAL or NORMAL_CONTROL for TETRA MOBILE system

or SYNC, NORMAL or NORMAL_SYNC for TETRA BASE and TETRA DIRECT systems

Note: NORMAL_CONTROL *and* NORMAL_SYNC *are not valid in the Power Analyzer display.*

Not applicable

Example : `:RECEiver:Bursttype NORMAL`
Sets the type of burst to be measured to normal bursts

:Bursttype?

Description : Returns the current burst type to be measured

Response : <CPD> CONTROL, NORMAL or NORMAL_CONTROL for TETRA MOBILE system

or SYNC, NORMAL or NORMAL_SYNC for TETRA BASE and TETRA DIRECT systems

Note: NORMAL_CONTROL *and* NORMAL_SYNC *are not valid in the Power Analyzer display.*

Example : `:RECEiver:Bursttype?`

Example response : `RECE:B NORMAL`

:RECEiver

:Channel

Description : Sets the frequency of the test set receiver by channel number

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:RECEiver:Channel 26`

:Channel?

Description : Returns the current test set receiver frequency channel number

Response : <NR1>

Example : `:RECEiver:Channel?`

Example response : `RECE:C 26`

:RECEiver

:Digautolev

Description : Sets the digital receiver autolevel state

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:RECEiver:Digautolev ON`
Switch digital receiver auto levelling ON

: Digautolev?

Description : Returns the setting of the receiver auto level function

Response : <CRD> OFF, ON

Example : `:RECEiver:Digautolev?`

Example response : `RECE:D ON`

:RECEiver

:FREQ

Description : Sets the frequency of the test set receiver

Parameters : <NRf>

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:RECEiver:FREQ 850.3MHZ`
Sets the test set receiver frequency to 850.3 MHz

:FREQ?

Description : Returns the current frequency setting of the test set receiver

Response : <NR2> Units in Hz

Example : `:RECEiver:FREQ?`

Example response : `RECE:FREQ 850300000HZ`

:RECEiver

:FRESn

Description : Sets the RF counter resolution to be used for transmitter RF frequency measurement.

Parameters : <CPD> RESN_1HZ
RESN_10HZ

Allowed suffices : Not applicable

Example : `:RECEiver:FRESn RESN_1HZ`
Set the RF counter resolution to 1 Hz

:FRESn?

Description : Returns the current setting of RF counter resolution

Response : <CRD> RESN_1HZ
RESN_10HZ

Example : `:RECEiver:FRESn?`

Example response : `RECE:FRES RESN_1HZ`

:RECEiver

:Ifband

Description : Selects the IF bandwidth filter to be used in the test set receiver.

Parameters : <CPD> DEFAULT
IFBAND_300HZ
IFBAND_1KHZ
IFBAND_2_4KHZ
IFBAND_3KHZ
IFBAND_7_5KHZ
IFBAND_10KHZ
IFBAND_15KHZ
IFBAND_30KHZ
IFBAND_110KHZ
IFBAND_280KHZ
IFBAND_3MHZ

Allowed suffices : Not applicable

Example : `:RECEiver:Ifband IFBAND_300HZ`
Select the 300 Hz IF bandwidth filter

:Ifband?

Description : Returns the current selection of test set receiver IF bandwidth filter

Response : <CRD> DEFAULT
IFBAND_300HZ
IFBAND_1KHZ
IFBAND_2_4KHZ
IFBAND_3KHZ
IFBAND_7_5KHZ
IFBAND_10KHZ
IFBAND_15KHZ
IFBAND_30KHZ
IFBAND_110KHZ
IFBAND_280KHZ
IFBAND_3MHZ

Example : `:RECEiver:Ifband?`

Example response : `RECE:I IFBAND_300HZ`

:RECEiver

:Powerbw

Description : Selects the power meter type for RF power measurement

Parameters : <CPD> BROADBAND, INBAND

Allowed suffices : Not applicable

Example : `:RECEiver:Powerbw BROADBAND`
Select the broad-band power meter

:Powerbw?

Description : Returns the current power meter selection

Response : <CRD> BROADBAND, INBAND

Example : `:RECEiver:Powerbw?`

Example response : `RECE:P BROADBAND`

:RESponse

Note : This command controls the way response data is returned to the controlling device, therefore it relates to the interpretation of the information given in this listing. See the note regarding this command at the beginning of this *Instrument-specific commands* section (page 5-30).

:Header

Description : Sets the response formatter output format.

Parameters : <CPD> OFF
MINIMUM
FULL
DEFAULT

Allowed suffices : Not applicable

Example : `:RESponse:Header MINIMUM`
Set response data headers to minimum

:Header?

Description : Returns the current setting of the response formatter output.

Response : <CRD> OFF
MINIMUM
FULL

Example : `:RESponse:Header?`

Example response : `OFF` (with OFF selected)

`RES:H MINIMUM` (with MINIMUM or DEFAULT selected)

`RESPONSE:HEADER FULL` (with FULL selected)

:RFGen

:Channel

Description : Sets the output frequency of the test set RF generator by channel number

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:RFGen:Channel 26`
Sets output frequency of the test set generator to channel 26

:Channel?

Description : Returns the current setting of the output frequency of the test set RF generator by channel number

Response : <NR1>

Example : `:RFGen:Channel?`

Example response : `RFG:C 26`

:RFGen

:Freq

Description : Sets the output frequency of the test set RF generator

Parameters : <NRf>

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:RFGen:Freq 543.21MHZ`
Sets the output frequency to 543.21 MHz

:Freq?

Description : Returns the current setting of the test set RF generator

Response : <NR2> Units in Hz

Example : `:RFGen:Freq?`

Example response : `RFG:F 543210000HZ`

:RFGen

:Level

Description : Sets the output level of the test set RF generator

Parameters : <NRf>

Allowed suffices : DBM

Example : `:RFGen:Level -80DBM`
Sets the RF generator output level to –80 dBm

:Level?

Description : Returns the current setting of the RF generator output level

Response : <NR2> Units in dBm

Example : `:RFGen:Level?`

Example response : `RFG:L -80DBM`

:RFGen

:Mode

Description : Sets the type of levelling used by the RF generator. Seamless mode allows a range of at least 20 dB without switching attenuators

Parameters : <CPD> NORMAL
SEAMLESS

Allowed suffices : Not applicable

Example : `:RFGen:Mode SEAMLESS`
Sets the RF generator mode to seamless

:Mode?

Description : Returns the type of levelling used by the RF generator

Response : <CRD> NORMAL
SEAMLESS

Example : `:RFGen:Mode?`

Example response : `RFG:M SEAMLESS`

:RFGen

:Status

Description : Sets the Status of the test set RF generator

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:RFGen:Status ON`
Sets Status of RF generator to ON

:Status?

Description : Returns the current status of the test set RF generator

Response : <CRD> OFF, ON

Example : `:RFGen:Status?`

Example response : `RFG:S ON`

:RFGen

:Topseamlevel

Description : Sets the level below which there is a range of at least 20 dB in seamless mode (see also RFGen:Mode)

Parameters : <NRf>

Allowed suffices : DBM

Example : `:RFGen:Topseamlevel -30DBM`
Sets the top limit of the 20 dB seamless range to –30 dB

:Topseamlevel?

Description : Returns the top limit of the 20 dB seamless range

Response : <NR2>

Example : `:RFGen:Topseamlevel?`

Example response : `RFG:T -30DBM`

:RFInput

Description : Sets the RF port input and output configuration

Parameters : <CPD> TNC_DUPLEX
N_DUPLEX
N_OUT_TNC_IN_DUPLEX
TNC_OUT_N_IN_DUPLEX
N_OUT_SIMPLEX
TNC_OUT_SIMPLEX
N_IN_SIMPLEX
TNC_IN_SIMPLEX
AUDIO_ONLY

Allowed suffices : Not applicable

Example : `:RFInput N_OUT_TNC_IN_DUPLEX`
Configure the N-type connector as the RF output port and the TNC connector as the RF input port for duplex operation

:RFInput?

Description : Returns the RF port input and output configuration

Response : <CRD> TNC_DUPLEX
N_DUPLEX
N_OUT_TNC_IN_DUPLEX
TNC_OUT_N_IN_DUPLEX
N_OUT_SIMPLEX
TNC_OUT_SIMPLEX
N_IN_SIMPLEX
TNC_IN_SIMPLEX
AUDIO_ONLY

Example : `:RFInput?`

Example response : `RFI N_OUT_TNC_IN_DUPLEX`

:RFMode

Description : Sets the type of RF TEST screen when already in RF TEST mode.

Parameters : <CPD> MENU
NORMAL
RF_TONES_SEQ_GEN
RF_TONES_SEQ_DEC
RF_TONES_CTCS
SRF_TONES_DCS
RF_TONES_DTMF
RF_TONES_DTMF_GEN
RF_TONES_DTMF_DEC
ACC_DPOWER
TXSSB
RXSSB

Allowed suffices : Not applicable

Example : `:RFMode RF_TONES_SEQ_GEN`
Sets the instrument to the SEQUENTIAL GENERATE RF screen when in RF TEST mode

:RFMode?

Description : Returns the current selection of RF TEST screen

Response : <CRD> MENU
NORMAL
RF_TONES_SEQ_GEN
RF_TONES_SEQ_DEC
RF_TONES_CTCS
SRF_TONES_DCS
RF_TONES_DTMF
RF_TONES_DTMF_GEN
RF_TONES_DTMF_DEC
ACC_DPOWER
TXSSB
RXSSB

Example : `:RFMode?`

Example response : `RFM RF_TONES_SEQ_GEN`

:RFPort

Description : Sets the RF port input and output configuration

Parameters : <CPD> TNC_DUPLEX
N_DUPLEX
N_OUT_TNC_IN_DUPLEX
TNC_OUT_N_IN_DUPLEX
N_OUT_SIMPLEX
TNC_OUT_SIMPLEX
N_IN_SIMPLEX
TNC_IN_SIMPLEX
AUDIO_ONLY

Allowed suffices : Not applicable

Example : `:RFPort N_OUT_TNC_IN_DUPLEX`
Configure the N-type connector as the RF output port and the TNC connector as the RF input port for duplex operation

:RFPort?

Description : Returns the RF port input and output configuration

Response : <CRD> TNC_DUPLEX
N_DUPLEX
N_OUT_TNC_IN_DUPLEX
TNC_OUT_N_IN_DUPLEX
N_OUT_SIMPLEX
TNC_OUT_SIMPLEX
N_IN_SIMPLEX
TNC_IN_SIMPLEX
AUDIO_ONLY

Example : `:RFPort`

Example response : `RFP N_OUT_TNC_IN_DUPLEX`

:RXDIstn

Note : The slaving of distortion measurements (both Rx and Tx) is dependent upon both the current mode and the slave setting, made either locally in the HELP AND SET-UP screens, or via remote control using the RXDIstn and TXDIStn commands. It will not be described here, as full details can be found in the Operating Manual 46882-274T.

:AFSlave

Description : Sets Rx distortion to be measured at either the user frequency, set with AFUser, or at the same frequency as AF generator 1

Parameters : <CPD> GEN1
USERFREQ

Allowed suffices : Not applicable

Example : `:RXDIstn:AFSlave GEN1`
Sets Rx distortion measurements to be done at the same frequency as AF generator 1.

:AFSlave?

Description : Returns the selected source of Rx distortion measuring frequency

Response : <CRD> GEN1
USERFREQ

Example : `:RXDIstn:AFSare?`

Example response : `RXDI:AFS GEN1`

:RXDIstn

Note : The slaving of distortion measurements (both Rx and Tx) is dependent upon both the current mode and the slave setting, made either locally in the HELP AND SET-UP screens, or via remote control using the RXDIstn and TXDIStn commands. It will not be described here, as full details can be found in the Operating Manual 46882-274T.

:AFUser

Description : Sets the user-defined AF frequency at which Rx distortion measurements are taken

Parameters : <NRf>

Allowed suffices : `HZ, KHZ`

Example : `:RXDIstn:AFUser 2.5KHZ`
Set the user-defined distortion measuring frequency to 2.5 kHz

:AFUser?

Description : Returns the current setting of user-defined distortion frequency

Response : <NR2> Frequency in Hz

Example : `:RXDIstn:AFUser?`

Example response : `RXDI:AFU 2500HZ`

:RXDIstn

Note : The slaving of distortion measurements (both Rx and Tx) is dependent upon both the current mode and the slave setting, made either locally in the HELP AND SET-UP screens, or via remote control using the RXDIstn and TXDIStn commands. It will not be described here, as full details can be found in the Operating Manual 46882-274T.

:MODSlave

Description : Sets Rx distortion to be measured at either the user-defined frequency, set with MODUser, or at the same frequency as modulation generator 1

Parameters : <CPD> GEN1
USERFREQ

Allowed suffices : Not applicable

Example : `:RXDIstn:MODSlave USERFREQ`
Sets frequency source for Rx distortion to user-defined frequency value.

:MODSlave?

Description : Returns the selected source of the modulation distortion measuring frequency

Response : <CRD> GEN1
USERFREQ

Example : `:RXDIstn:MODSlave`

Example response : `RXDI:MODS USERFREQ`

:RXDIstn

Note : The slaving of distortion measurements (both Rx and Tx) is dependent upon both the current mode and the slave setting, made either locally in the HELP AND SET-UP screens, or via remote control using the RXDIstn and TXDIStn commands. It will not be described here, as full details can be found in the Operating Manual 46882-274T.

:MODUser

Description : Sets the user-defined frequency at which Rx distortion measurements are taken.

Parameters : <NRf>

Allowed suffices : `HZ, KHZ`

Example : `:RXDIstn:MODUser 2.5KHZ`
Set the user-defined distortion measuring frequency to 2.5 kHz

:MODUser?

Description : Returns the current setting of user-defined distortion measuring frequency.

Response : <NR2> Frequency in Hz

Example : `:RXDIstn:MODUser?`

Example response : `RXDI:MODU 2500HZ`

:RXDType

Description : Sets the audio distortion measurement type in Rx TEST mode.

Parameters : <CPD> OFF
DISTN
SINAD
SN_RATIO

Allowed suffices : Not applicable

Example : `:RXDType DISTN`
Select Distortion as the measurement mode

:RXDType?

Description : Returns the current distortion measurement type setting

Response : <CRD> OFF
DISTN
SINAD
SN_RATIO

Example : `:RXDType?`

Example response : `RXDT DISTN`

:RXEqtx

Description : Actions the Rx=Tx function. Sets the RF generator to the same frequency as the received frequency. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:RXEqtx`

:RXFilt

Description : Sets the type of audio frequency filter

Parameters : <CPD> OFF
LP_5KHZ
LP_20KHZ
LP_300HZ
STD_BP
PSOPH

Allowed suffices : Not applicable

Example : `:RXFilt LP_20KHZ`
Select the 20 kHz low-pass filter

:RXFilt?

Description : Returns the current setting of Rx TEST AF filter

Response : <CRD> OFF
LP_5KHZ
LP_20KHZ
LP_300HZ
STD_BP
PSOPH

Example : `:RXFilt?`

Example response : `RXF LP_20KHZ`

:RXMode

Description : Sets the type of Rx TEST screen, when already in Rx TEST mode.

Response : <CPD> NORMAL
RX_TONES_SEQ_GEN
RX_TONES_CTCSS
RX_TONES_DTMF
RX_TONES_DTMF_GEN
RX_TONES_DTMF_DEC
RX_TONES_DCS
RX_TONES_POCSAG

Example : `Not applicable`

Example response : `RXM RX_TONES_POCSAG`
Sets the instrument to the POCSAG test screen when in RX test mode.

:RXMode?

Description : Returns the current setting of the Rx TEST mode screen.

Response : <CRD> NORMAL
RX_TONES_SEQ_GEN
RX_TONES_CTCSS
RX_TONES_DTMF
RX_TONES_DTMF_GEN
RX_TONES_DTMF_DEC
RX_TONES_DCS
RX_TONES_POCSAG

Example : `:RXMode?`

Example response : `RXM RX_TONES_POCSAG`

:SBandana

:MARkers

Description : Sets the status of the sideband analyzer markers

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:SBandana:MARkers ON`
Sets sideband analyzer markers to ON

:MARkers?

Description : Returns the current status of the sideband analyzer markers

Response : <CRD> OFF, ON

Example : `:SBandana:MARkers?`

Example response : `SB:MAR ON`

:SBandana

:MAXhold

Description : Sets, releases, or resets, the Maximum Hold function on the sideband analyzer

Parameters : <CPD> OFF
ON

Allowed suffices : Not applicable

Example : `:SBandana:MAXhold ON`
Sets Maximum Hold function

:MAXhold?

Description : Returns the current setting of the Maximum Hold function

Response : <CRD> OFF
ON

Example : `:SBandana:MAXhold?`

Example response : `SB:MAX ON`

:SBandana

:MKR*n*

Description : Sets the frequency location of sideband analyzer marker *n* where *n* = 1 or 2

Parameters : <NRf>

Allowed suffices : `HZ, KHZ`

Example : `:SBandana:MKR1 2.5KHZ`
Set marker 1 to 2.5 kHz

:MKR*n*?

Description : Returns the current frequency location of sideband analyzer marker *n* where *n* = 1 or 2

Response : <NR2> Frequency in Hz

Example : `:SBandana:MKR1?`

Example response : `SB:MKR1 2500HZ`

:SBandana

:Peakfind

Description : Causes the sideband analyzer marker 1 to locate at the point of maximum signal level. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:SBandana:Peakfind`
Locates sideband analyzer marker 1 onto highest signal level point

:SBandana

:Reflevel

Description : Sets the sideband analyzer reference level relative to the top of screen.

Parameters : <NRf>

Allowed suffices : DB

Example : `:SBandana:Reflevel 20DB`
Set the sideband analyzer reference level to 20 dB above the top of screen

:Reflevel?

Description : Returns the current setting of the sideband analyzer reference level

Response : <NR2> Unit in dB

Example : `:SBandana:Reflevel?`

Example response : `SB:REF 20DB`

:SBandana

:SIze

Description : Sets the view of the sideband analyzer mode to normal or expanded.

Parameters : <CPD> NORMAL, EXPAND

Allowed suffices : Not applicable

Example : `:SBandana:SIze EXPAND`
Set the sideband analyzer display to expanded view

:SIze?

Description : Returns the current setting of the sideband analyzer display view

Response : <CRD> NORMAL, EXPAND

Example : `:SBandana:SIze?`

Example response : `SB:SI EXPAND`

:SBandana

:SPan

Description : Selects the side band analyzer sweep span/division

Parameters : <CPD> AS_5HZ
AS_10HZ
AS_25HZ
AS_50HZ
AS_100HZ
AS_250HZ
AS_500HZ

Allowed suffices : Not applicable

Example : `:SBandana:SPan AS_5HZ`
Set the sideband analyzer sweep span to 5 Hz/division

:SPan?

Description : Returns the current setting of the sideband analyzer sweep span

Response : <CRD> AS_5HZ
AS_10HZ
AS_25HZ
AS_50HZ
AS_100HZ
AS_250HZ
AS_500HZ

Example : `:SBandana:SPan?`

Example response : `SB:SP AS_5HZ`

:SBandana

:Trig

Description : Selects the sideband analyzer sweep trigger mode.

Parameters : <CPD> SINGLE, REPEAT

Allowed suffices : Not applicable

Example : `:SBandana:Trig REPEAT`
Set the sideband analyzer sweep trigger mode to repeat.

:Trig?

Description : Returns the current setting of the sideband analyzer trigger mode.

Response : <CRD> SINGLE, REPEAT

Example : `:SBandana:Trig?`

Example response : `SB:T REPEAT`

:SBandana

:Vrange

Description : Sets the sideband analyzer vertical range by dB per division

Parameters : <NRf>

Allowed suffices : DB

Example : `:SBandana:Vrange 5DB`
Set the vertical range to 5 dB/division

:Vrange?

Description : Returns the current setting of the sideband analyzer vertical range

Response : <NR2> Units in dB

Example : `:SBandana:Vrange?`

Example response : `SB:V 5DB`

:SCope

:AFrange

Description : Sets the AF vertical span/div of the oscilloscope

Parameters : <CPD> SC_2MV
SC_5MV
SC_10MV
SC_20MV
SC_50MV
SC_100MV
SC_200MV
SC_500MV
SC_1V
SC_2V
SC_5V
SC_10V
SC_20V

Allowed suffices : Not applicable

Example : `:SCope:AFrange SC_2MV`
Set the oscilloscope AF vertical range to 2 mV/division

:AFrange?

Description : Returns the current setting of the oscilloscope AF vertical span

Response : <CRD> SC_2MV
SC_5MV
SC_10MV
SC_20MV
SC_50MV
SC_100MV
SC_200MV
SC_500MV
SC_1V
SC_2V
SC_5V
SC_10V
SC_20V

Example : `:SCope:AFrange?`

Example response : `SC:AF SC_2MV`

:SCope

:AMrange

Description : Sets the AM vertical span of the oscilloscope

Parameters : <CPD> SC_5PC
SC_10PC
SC_20PC

Allowed suffices : Not applicable

Example : `:SCope:AMrange SC_5PC`
Set the oscilloscope AM range to 5%/division

:AMrange?

Description : Returns the current setting of the oscilloscope AM vertical span

Response : <CRD> SC_5PC
SC_10PC
SC_20PC

Example : `:SCope:AMrange?`

Example response : `SC:AM SC_5PC`

:SCope

:Fmrange

Description : Sets the FM vertical span of the oscilloscope

Parameters : <CPD> SC_500HZ
SC_1KHZ
SC_2KHZ
SC_5KHZ
SC_10KHZ
SC_20KHZ
SC_50KHZ

Allowed suffices : Not applicable

Example : `:SCope:Fmrange SC_200HZ`
Sets the oscilloscope FM range to 200 Hz/division

:Fmrange?

Description : Returns the current setting of the oscilloscope FM vertical span

Response : <CRD> SC_500HZ
SC_1KHZ
SC_2KHZ
SC_5KHZ
SC_10KHZ
SC_20KHZ
SC_50KHZ

Example : `:SCope:Fmrange?`

Example response : `SC:F SC_200HZ`

:SCope

:MArkers

Description : Sets the status of the oscilloscope markers

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:SCope:MArkers ON`
Set the oscilloscope markers ON

:MArkers?

Description : Returns the current status of the oscilloscope markers

Response : <CRD> OFF, ON

Example : `:SCope:MArkers?`

Example response : `SC:MA ON`

:SCope

:MKR*n*

Description : Sets the horizontal location of oscilloscope marker *n* relative to 0 s (left side of graticule) where *n* = 1 or 2

Parameters : <NRf>

Allowed suffices : `US, MS, S`

Example : `:SCope:MKR1 20MS`
Locate oscilloscope marker 1, 20 ms from 0 s

:MKR*n*?

Description : Returns the current horizontal location of oscilloscope marker *n* where *n* = 1 or 2

Response : <NR2> Time in µsecs

Example : `:SCope:MKR1?`

Example response : `SC:MKR1 2000US`

:SCope

:PMrange

Description : Sets the PM vertical span of the oscilloscope

Parameters : <CPD> SC_0_5RAD
SC_1RAD
SC_2RAD
SC_5RAD

Allowed suffices : Not applicable

Example : `:SCope:PMrange SC_0_5RAD`
Sets the oscilloscope PM range to 5 radians/division

:PMrange?

Description : Returns the current setting of the oscilloscope PM vertical span

Response : <CRD> SC_0_5RAD
SC_1RAD
SC_2RAD
SC_5RAD

Example : `:SCope:PMrange?`

Example response : `SC:PM SC_0_5RAD`

:SCope

Note : The SCOPE POSITION control on the front panel remains operative when the instrument is under remote control. Therefore the value set may not always be the value returned by a query.

:POsItion

Description : Sets the vertical position of the oscilloscope trace

Parameters : <NR1> –128 to 127. 0 is centred for 0 V DC coupled.

Allowed suffices : Not applicable

Example : `:SCope:POsition 50`

:POsItion?

Description : Returns the current vertical position of the oscillocope trace

Response : <NR1>

Example : `:SCope:POsition?`

Example response : `SC:PO 50`

:SCope

:SIze

Description : Sets the view of the oscilloscope display to normal or expanded

Parameters : <CPD> NORMAL, EXPAND

Allowed suffices : Not applicable

Example : `:SCope:SIze EXPAND`
Set the oscilloscope display to expanded view

:SIze?

Description : Returns the current setting of the oscilloscope display view

Response : <CRD> NORMAL, EXPAND

Example : `:SCope:SIze?`

Example response : `SC:SI EXPAND`

:SCope

:SPan *or* :TBase

Description : Sets the horizontal span/div of the oscilloscope

Parameters : <CPD> SC_5US
SC_10US
SC_20US
SC_50US
SC_100US
SC_200US
SC_500US
SC_1MS
SC_2MS
SC_5MS
SC_10MS
SC_20MS
SC_50MS
SC_100MS
SC_200MS
SC_500MS
SC_1S
SC_2S
SC_5S
SC_10S

Allowed suffices : Not applicable

Examples : `:SCope:TBase SC_5US`
`:SCope:SPan SC_5US`
Set the oscilloscope timebase to 5 µs/division

:SPan? *or* :TBase?

Description : Returns the current setting of the oscilloscope horizontal span

Response : <CRD> SC_5US
SC_10US
SC_20US
SC_50US
SC_100US
SC_200US
SC_500US
SC_1MS
SC_2MS
SC_5MS
SC_10MS
SC_20MS
SC_50MS
SC_100MS
SC_200MS
SC_500MS
SC_1S
SC_2S
SC_5S
SC_10S

Examples :
```
:SCope:TBase?
:SCope:SPan?
```

Example response :
```
SC_5US
```

:SCope

:TRig

Description : Controls the oscilloscope trigger mode.

Parameters : <CPD> SINGLE, REPEAT

Allowed suffices : Not applicable

Example :
```
:SCope:TRig REPEAT
```
Set the oscilloscope trigger mode to repeat

:TRig?

Description : Returns the current setting of the oscilloscope trigger mode

Response : <CRD> SINGLE, REPEAT

Example :
```
:SCope:TRig?
```

Example response :
```
SC:TR REPEAT
```

:SPecana

:Filter

Description : Controls the spectrum analyzer resolution bandwidth filter selection

Parameters : <CPD> AUTO
SAFIL_300HZ
SAFIL_1KHZ
SAFIL_3KHZ
SAFIL_10KHZ
SAFIL_30KHZ
SAFIL_100KHZ
SAFIL_300KHZ
SAFIL_3MHZ

Allowed suffices : Not applicable

Example : `:SPecana:Filter SAFIL_300HZ`
The resolution bandwidth of the spectrum analyzer is 300 Hz

:Filter?

Description : Returns the current setting of the spectrum analyzer resolution bandwidth

Response : <CRD> SAFIL_300HZ
SAFIL_1KHZ
SAFIL_3KHZ
SAFIL_10KHZ
SAFIL_30KHZ
SAFIL_100KHZ
SAFIL_300KHZ
SAFIL_3MHZ

Example : `:SPecana:Filter?`

Example response : `SP:F SAFIL_300HZ`

:SPecana

:MARkers

Description : Sets the status of the spectrum analyzer markers

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:SPecana:MARkers ON`
Set the status of the spectrum analyzer markers to ON

:MARkers?

Description : Returns the current status of the spectrum analyzer markers

Response : <CRD> OFF, ON

Example : `:SPecana:MARkers?`

Example response : `SP:MAR ON`

:SPecana

:MAXhold

Description : Sets, releases, or resets, the Maximum Hold function on the spectrum analyzer

Parameters : <CPD> OFF
ON

Allowed suffices : Not applicable

Example : `:SPecana:MAXhold ON`
Sets Maximum Hold function

:MAXhold?

Description : Returns the current setting of the Maximum Hold function

Response : <CRD> OFF
ON

Example : `:SPecana:MAXhold?`

Example response : `SP:MAX ON`

:SPecana

:MKR*n*

Description : Sets the horizontal (frequency) location of spectrum analyzer marker *n*, where *n* = 1 or 2.

Parameters : <NRf>

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:SPecana:MKR1 435MHZ`
Set spectrum analyzer marker 1 to 435 MHz

:MKR*n*?

Description : Returns the current horizontal (frequency) location of spectrum analyzer marker *n*, where *n* = 1 or 2

Response : <NR2> Frequency in Hz

Example : `:SPecana:MKR1?`

Example response : `SP:MKR1 435000000HZ`

:SPecana

:Peakfind

Description : Locates spectrum analyzer marker 1 at the point of maximum signal level. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:SPecana:Peakfind`

:SPecana

:REFFreq

Description : Sets the reference frequency of the spectrum analyzer

Parameters : <NRf>

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:SPecana:REFFreq 250MHZ`
Sets the reference frequency of the spectrum analyzer to 250 MHz

:REFFreq?

Description : Returns the current setting of the spectrum analyzer reference frequency

Response : <NR2> Frequency in Hz

Example : `:SPecana:REFFreq?`

Example response : `SP:REFF 250000000HZ`

:SPecana

:REFLevel

Description : Sets the reference level of the spectrum analyzer

Parameters : <NRf>

Allowed suffices : `DBM`

Example : `:SPecana:REFLevel -45DBM`
Set the spectrum analyzer reference level to –45 dBm

:REFLevel?

Description : Returns the current setting of the spectrum analyzer reference level

Response : <NR2> Units in dBm

Example : `:SPecana:REFLevel?`

Example response : `SP:REFL -45DBM`

:SPecana

:SIze

Description : Sets the view of the spectrum analyzer to normal or expanded

Parameters : <CPD> NORMAL, EXPAND

Allowed suffices : Not applicable

Example : `:SPecana:SIze EXPAND`
Sets the spectrum analyzer diplay to expanded view

:SIze?

Description : Returns the current setting of the spectrum analyzer display size

Response : <CRD> NORMAL, EXPAND

Example : `:SPecana:SIze?`

Example response : `SP:SI EXPAND`

:SPecana

:SPan

Description : Sets the horizontal (frequency) span/division of the spectrum analyzer

Parameters : <NRf>

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:SPecana:SPan 10.3MHZ`
Set the horizontal span/division of the spectrum analyzer to 10.3 MHz/division

:SPan?

Description : Returns the current setting of the spectrum analyzer horizontal (frequency) span/division

Response : <NR2> Frequency in Hz

Example : `:SPecana:SPan?`

Example response : `SP:SP 10300000HZ`

:SPecana

:TRAck

Description : Sets the status of the spectrum analyzer tracking generator. This is analogous to TRackgen:Status.

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:SPecana:TRAck OFF`
Set the spectrum analyzer tracking generator status to OFF

:TRAck?

Description : Returns the current status of the spectrum analyzer tracking generator

Response : <CRD> OFF, ON

Example : `:SPecana:TRAck?`

Example response : `SP:TRA OFF`

:SPecana

:TRIg

Description : Controls the spectrum analyzer trigger mode selection

Parameters : <CPD> SINGLE, REPEAT

Allowed suffices : Not applicable

Example : `:SPecana:TRIg SINGLE`
Sets the spectrum analyzer sweep trigger mode to single sweep

:TRIg?

Description : Returns the current setting of the spectrum analyzer trigger mode

Response : <CRD> SINGLE, REPEAT

Example : `:SPecana:TRIg?`

Example response : `SP:TRI SINGLE`

:TIme

Description : Sets the time for the real-time clock

Parameters : <NR1>,<NR1>,<NR1> Hours, mins, and secs in format *hh, mm, ss*

Allowed suffices : Not applicable

Example : `:TIme 09,35,00`
Sets the real-time clock time to 0935

:TIme?

Description : Returns the current setting of the real-time clock time

Response : <NR1>, <NR1>, <NR1> *hh, mm, ss*

Example : `:TIme?`

Example response : `TI 09,35,00`

:TRackgen

:MIxerlo

Description : Sets the offset frequency of the spectrum analyzer tracking generator, for up or down mixer testing

Parameters : <NRf>

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:TRackgen:MIxerlo 45MHZ`
Set the tracking generator frequency offset to 45 MHz

:MIxerlo?

Description : Returns the current setting of offset frequency of the spectrum analyzer tracking generator

Response : <NR2> Frequency in Hz

Example : `:TR:MI?`

Example response : `TR:MI 45000000HZ`

:TRackgen

:MOde

Description : Sets the operating mode of the spectrum analyzer tracking generator

Parameters : <CPD> NORMAL
DOUBLER
DIVIDER
UP_MIXER
DOWN_MIXER

Allowed suffices : Not applicable

Example : `:TRackgen:MOde DOUBLER`
Set the spectrum analyzer tracking generator mode to doubler test

:MOde?

Description : Returns the current status of the spectrum analyzer tracking generator

Response : <CRD> NORMAL
DOUBLER
DIVIDER
UP_MIXER
DOWN_MIXER

Example : `:TRackgen:MOde?`

Example response : `TR:MO DOUBLER`

:TRackgen

:Status

Description : Sets the status of the spectrum analyzer tracking generator

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:TRackgen:Status OFF`
Set the spectrum analyzer tracking generator status to OFF

:Status?

Description : Returns the current status of the spectrum analyzer tracking generator

Response : <CRD> OFF, ON

Example : `:TRackgen:Status?`

Example response : `TR:S OFF`

:TXDIstn

Note : The slaving of distortion measurements (both Rx and Tx) is dependent upon both the current mode and the slave setting, made either locally in the HELP AND SET-UP screens, or via remote control using the RXDIstn and TXDIStn commands. It will not be described here, as full details can be found in the Operating Manual 46882-274T.

:AFSlave

Description : Sets Tx distortion to be measured at either the user frequency, set with AFUser or at the same frequency as AF generator 1.

Parameters : <CPD> GEN1, USERFREQ

Allowed suffices : Not applicable

Example : `:TXDIstn:AFSlave GEN1`
Sets Tx distortion measurements to be done at the same frequency as AF generator 1.

:AFSlave?

Description : Returns the selected source of Tx distortion measuring frequency.

Response : <CRD> GEN1, USERFREQ

Example : `:TXDIstn:AFSlave?`

Example response : `TXDI:AFS GEN1`

:TXDIstn

:AFUser

Description : Sets the user-defined AF frequency at which Tx distortion measurements are taken.

Parameters : <NRf>

Allowed suffices : `HZ, KHZ`

Example : `:TXDIstn:AFUser 2.5KHZ`
Set the user-defined distortion measuring frequency to 2.5 kHz.

:AFUser?

Description : Returns the current setting of user-defined distortion frequency.

Response : <NR2> Frequency in Hz

Example : `:TXDIstn:AFUser?`

Example response : `TXDI:AFU 2500HZ`

:TXDIstn

Note : The slaving of distortion measurements (both Rx and Tx) is dependent upon both the current mode and the slave setting, made either locally in the HELP AND SET-UP screens, or via remote control using the RXDIstn and TXDIStn commands. It will not be described here, as full details can be found in the Operating Manual 46882-274T.

:MODSlave

Description : Sets Tx distortion to be measured at either the user-defined frequency, set with :MODUser (page 5-174), or at the same frequency as Mod gen1.

Parameters : <CPD> GEN1, USERFREQ

Allowed suffices : Not applicable

Example : `:TXDIstn:MODSlave USERFREQ`
Sets frequency source for Tx distortion measurements to user-defined frequency value.

:MODSlave?

Description : Returns selected source of Tx distortion measuring frequency.

Response : <CRD> GEN1, USERFREQ

Example : `:TXDIstn:MODSlave?`

Example response : `TXDI:MODS USERFREQ`

:TXDIstn

Note : The slaving of distortion measurements (both Rx and Tx) is dependent upon both the current mode and the slave setting, made either locally in the HELP AND SET-UP screens, or via remote control using the RXDIstn and TXDIStn commands. It will not be described here, as full details can be found in the Operating Manual 46882-274T.

:MODUser

Description : Sets the user-defined modulation frequency at which Tx distortion measurements are taken.

Parameters : <NRf>

Allowed suffices : `HZ, KHZ`

Example : `:TXDIstn:MODUser 2.5KHZ`
Set the user-defined distortion measuring frequency to 2.5 kHz

:MODUser?

Description : Returns the current setting of user-defined distortion measuring frequency.

Response : <NR2> Frequency in Hz

Example : `:TXDIstn:MODUser?`

Example response : `TXDI:MODU 2500HZ`

:TXDType

Description : Sets the transmitter distortion measurement type

Parameters : <CPD> OFF
DISTN
SN_RATIO

Allowed suffices : Not applicable

Example : `:TXDType SN_RATIO`
Sets signal-to-noise ratio as the transmitter test distortion measuring method

:TXDType?

Description : Returns the current selection of distortion measurement type

Response : <CRD> OFF
DISTN
SN_RATIO

Example : `:TXDType?`

Example response : `TXDT SN_RATIO`

:TXFilt

Description : Sets the transmitter test demodulation filter passband

Parameters : <CPD> OFF
LP_20KHZ
LP_5KHZ
LP_300HZ
STD_BP
PSOPH

Allowed suffices : Not applicable

Example : `:TXFilt LP_20KHZ`
Set 20 kHz low pass as the transmitter test demodulation filter

:TXFilt?

Description : Returns the current setting of the transmitter test demodulation filter

Response : <CRD> OFF
LP_20KHZ
LP_5KHZ
LP_300HZ
STD_BP
PSOPH

Example : `:TXFilt?`

Example response : `TXF LP_20KHZ`

:TXMode

Description : Sets the type of Tx TEST screen, when already in a Tx TEST mode screen.

Parameters : <CPD> NORMAL
OFF_AIR
ACC_DPOWER
TX_TONES_SEQ
TX_TONES_DTMF
TX_TONES_DTMF_GEN
TX_TONES_DTMF_DEC

Allowed suffices : Not applicable

Example : `:TXMode OFF_AIR`
Set the instrument to the OFF AIR screen when in Tx TEST mode

:TXMode?

Description : Returns the current setting of Tx TEST mode

Response : <CRD> NORMAL
OFF_AIR
ACC_DPOWER
TX_TONES_SEQ
TX_TONES_DTMF
TX_TONES_DTMF_GEN
TX_TONES_DTMF_DEC

Example : `:TXMode?`

Example response : `TXM OFF_AIR`

:UNitmeas

:Aflevel

Description : Sets the unit in which AF measurements are returned. The relative level is reset whenever this command is received. AFL_DBM is valid only when the 600 Ω termination is connected.

Parameters : <CPD> AFL_VOLT
AFL_DBV
AFL_DBM
AFL_DBR

Allowed suffices : Not applicable

Example : `:UNitmeas:Aflevel AFL_VOLTS`
Sets the unit to be returned for AF Level to Volts

:Aflevel?

Description : Returns current value for AF Level unit

Response : <CRD> AFL_VOLT
AFL_DBV
AFL_DBM
AFL_DBR

Example : `:UNitmeas:Aflevel?`

Example response : `UN:A AFL_VOLT`

:UNitmeas

:Demodlevel

Description : Selects the type of demodulation measurement

Parameters : <CPD> PEAK
RMS

Allowed suffices : Not applicable

Example : `:UNitmeas:Demodlevel RMS`
Sets demodulation measurements to be RMS

:Demodlevel?

Description : Returns current type of demodulation measurement

Response : <CRD> PEAK
RMS

Example : `:UNitmeas:Demodlevel?`

Example response : `UN:D RMS`

:USeroptions

:Afpad

Description : Sets the status of the AF generator 20 dB external pad offset

Parameters : <CPD> OUT, IN

Allowed suffices : Not applicable

Example : `:USeroptions:Afpad OUT`
Set the AF generator 20 dB external offset to OUT (not connected)

:Afpad?

Description : Returns the current status of the AF generator 20 dB external pad offset

Response : <CPD> OUT, IN

Example : `:USeroptions:Afpad?`

Example response : `US:A OUT`

:USeroptions

:Freqstd

Description : Sets the source of the test set frequency standard.

Parameters : <CPD> INTERNAL, EXTERNAL

Allowed suffices : Not applicable

Example : `:USeroptions:Freqstd INTERNAL`
The instrument is using the internal frequency standard

:Freqstd?

Description : Returns the current setting of the instrument frequency standard source

Response : <CRD> INTERNAL, EXTERNAL

Example : `:USeroptions:Freqstd?`

Example response : `US:F INTERNAL`

:USeroptions

:Genmode

Description : Sets the AF generator or modulation generator mode to one 100 kHz generator or six 20 kHz generators

Parameters : <CPD> DEFAULT_6_GENS
SINGLE_100K_GEN

Allowed suffices : Not applicable

Example : `:USeroptions:Genmode SINGLE_100K_GEN`
Set the AF generator or modulation generator mode to one 100 kHz generator

:Genmode?

Description : Returns the current mode of the AF generator or modulation generator mode

Response : <CRD> DEFAULT_6_GENS
SINGLE_100K_GEN

Example : `:USeroptions:Genmode?`

Example response : `US:G SINGLE_100K_GEN`

:USeroptions

:Inlevoffset

Description : Sets the value of external amplifier offset in the RF input

Parameters : <NRf>

Allowed suffices : `DB`

Example : `:USeroptions:Inlevoffset 20DB`
The RF input level offset is 20 dB

:Inlevoffset?

Description : Returns the current value of external amplifier offset in the RF input

Response : <NR2> Units of dB

Example : `:USeroptions:Inlevoffset?`

Example response : `US:I 20DB`

:USeroptions

:Language

Description : Sets the language used on the display

Parameters : <CPD> ENGLISH
FRENCH
GERMAN
SPANISH

Allowed suffices : Not applicable

Example : `:USeroptions:Language ENGLISH`
Set English as the language used on the display

:Language?

Description : Returns the current setting of display language

Response : <CRD> ENGLISH
FRENCH
GERMAN
SPANISH

Example : `:USeroptions:Language?`

Example response : `US:L ENGLISH`

:USeroptions

:PRINTPort

Description : Sets which port is used for printer output

Parameters : <CPD> PARALLEL, SERIAL

Allowed suffices : Not applicable

Example : `:USeroptions:PRINTPort PARALLEL`
Sets the parallel port as the printer output port

:PRINTPort?

Description : Returns the current selection of port for printer output

Response : <CRD> PARALLEL, SERIAL

Example : `:USeroptions:PRINTPort?`

Example response : `US:PRINTP PARALLEL`

:USeroptions

:PRINTType

Description : Sets the driver for the printer output

Parameters : <CPD> EPSON80
EPSON100
LASER100
LASER150
LASER300

Allowed suffices : Not applicable

Example : `:USeroptions:PRINTType EPSON80`
The printer driver is set for an EPSON 80 printer

:PRINTType?

Description : Returns the current setting of printer driver

Response : <CRD> EPSON80
EPSON100
LASER100
LASER150
LASER300

Example : `:USeroptions:PRINTType?`

Example response : `US:PRINTT EPSON80`

:USeroptions

:PSoph_select

Description : Sets the type of psophometric filter

Parameters : <CPD> CCITT, CMESS

Allowed suffices : Not applicable

Example : `:USeroptions:PSoph_select CCITT`
Selects CCITT to be the psophometric filter type.

:PSoph_select?

Description : Returns the current selection of psophometric filter

Response : <CRD> CCITT, CMESS

Example : `:USeroptions:PSoph_select?`

Example response : `US:PS CCITT`

:USeroptions

:REMOTEIn

Description : Sets which port is used for remote control input

Parameters : <CPD> GPIB
SERIAL
SERIAL _OR_GPIB

Allowed suffices : Not applicable

Example : `:USeroptions:REMOTEIn GPIB`
Receive remote control input from the GPIB port

:REMOTEIn?

Description : Returns the current selection of port for remote control input

Response : <CRD> GPIB
SERIAL
SERIAL _OR_GPIB

Example : `:USeroptions:REMOTEIn?`

Example response : `US:REMOTEI GPIB`

:USeroptions

:REMOTEOut

Description : Sets which port is used for remote control output

Parameters : <CPD> GPIB, SERIAL

Allowed suffices : Not applicable

Example : `:USeroptions:REMOTEOut GPIB`
Send remote control output from the GPIB port

:REMOTEOut?

Description : Returns the current selection of port for remote control output

Response : <CRD> GPIB, SERIAL

Example : `:USeroptions:REMOTEOut?`

Example response : `US:REMOTEO GPIB`

:USeroptions

:RFlevoffset

Description : Sets the value of external amplifier offset in the RF generator output

Parameters : <NRf>

Allowed suffices : DB

Example : `:USeroptions:RFlevoffset 20DB`
External level offset in RF generator output is 20 dB

:RFlevoffset?

Description : Returns the current value of external amplifier offset in the RF generator output

Response : <NR2> Units of dB

Example : `:USeroptions:RFlevoffset?`

Example response : `US:RF 20DB`

:USeroptions

:Stdfreq

Description : Sets which of the frequency standards is to be used by the instrument. The source may be internal or external, as set by the USeroptions:Freqstd command.

Parameters : <CPD> STDFREQ_10MHZ
STDFREQ_13MHZ

Allowed suffices : Not applicable

Example : `:USeroptions:Stdfreq STDFREQ_13MHZ`
Sets the frequency (internal or external) to be used by the instrument to 13 MHz

:Stdfreq?

Description : Returns the internal or external frequency standard being used by the instrument

Response : <CRD> STDFREQ_10MHZ
STDFREQ_13MHZ

Example : `:USeroptions:Stdfreq?`

Example response : `US:S STDFREQ_13MHZ`

:USeroptions

:Termaf

Description : Sets the state of the AF generator termination impedance for use with a balanced output unit

Parameters : <CPD> TERM_1M
TERM_600

Allowed suffices : Not applicable

Example : `:USeroptions:Termaf TERM_600`
Sets the AF generator termination impedance to 600 Ω

:Termaf?

Description : Returns the state of the AF generator termination impedance

Response : <CRD> TERM_1M
TERM_600

Example : `:USeroptions:Termaf?`

Example response : `US:T TERM_600`

:VECTDiag

:Refresh

Description : Set the vector diagram display refresh method to continual refresh, accumulation or with persistence.

Parameters : <CPD> REFRESH
ACCUMULATE
PERSISTENCE (invalid for phase trajectory display)

Allowed suffices : Not applicable

Example : `:VECTDiag:Refresh ACCUMULATE`
Set the vector diagram display refresh method to accumulate

:Refresh?

Description : Returns the current setting of the vector diagram display refresh method.

Response : <CRD> REFRESH
ACCUMULATE
PERSISTENCE

Example : `:VECTDiag:Refresh?`

Example response : `VECTD:R ACCUMULATE`

:VECTDiag

:Size

Description : Set the size of the vector diagram to normal or expanded.

Parameters : <CPD> NORMAL
EXPAND

Allowed suffices : Not applicable

Example : `:VECTDiag:Size EXPAND`
Set the vector diagram to expanded view

:Size?

Description : Returns the current setting of the vector diagram size.

Response : <CRD> NORMAL
EXPAND

Example : `:VECTDiag:Size?`

Example response : `VECTD:S EXPAND`

:VECTDiag

:TRig

Description : Set the vector diagram trigger mode

Parameters : <CPD> SINGLE
REPEAT

Allowed suffices : Not applicable

Example : `:VECTDiag:TRig REPEAT`
Set the vector diagram trigger mode to repeat

:TRig?

Description : Returns the current setting of the vector diagram trigger mode

Response : <CRD> SINGLE
REPEAT

Example : `: VECTDiag:TRig?`

Example response : `VECTD:TR REPEAT`

:VECTDiag

:TYpe

Description : .Set the type of display for the vector diagram

Parameters : <CPD> CONSTELLATION
TRAJECTORY
ROTATED

Allowed suffices : Not applicable

Example : `:VECTDiag:TYpe CONSTELLATION`
Set the power analyser display to symbol constellations

:TYpe?

Description : Returns the current setting of the vector diagram type

Response : <CRD> CONSTELLATION
TRAJECTORY
ROTATED

Example : `:VECTDiag:TYpe?`

Example response : `VECTD:TY ROTATED`

:VOlume

Note : The VOLUME control on the front panel remains operative when the instrument is under remote control. Therefore the value set may not always be the value returned by a query.

Description : Sets the volume of the test set loudspeaker

Parameters : <NR1> Range 0 to 255

Allowed suffices : Not applicable

Example : `:VOlume 50`
Sets volume level to 50 arbitrary units

:VOlume?

Description : Returns the current level of the test set volume setting

Response : <NR1>

Example : `:VOlume?`

Example response : *V 60*

Tones operation commands

List of commands

:DCstones
- :AMDECodelevel
- :AMDECodelevel?
- :AMDEPth
- :AMDEPth?
- :Bitrate
- :Bitrate?
- :Code
- :Code?
- :FMDECodelevel
- :FMDECodelevel?
- :FMDEVn
- :FMDEVn?
- :Mode
- :Mode?
- :PMDECodelevel
- :PMDECodelevel?
- :PMDEVn
- :PMDEVn?
- :REadcode?
- :RXpolarity
- :RXpolarity?
- :Txpolarity
- :Txpolarity?

:DTmftones
- :AFdecodelevel
- :AFdecodelevel?
- :AMdecodelevel
- :AMdecodelevel?
- :DEcodereset
- :DUration
- :DUration?
- :Fmdecodelevel
- :Fmdecodelevel?
- :HIAFlevel
- :HIAFlevel?
- :HIAMdepth
- :HIAMdepth?
- :HIFMdevn
- :HIFMdevn?
- :HIFReqshift
- :HIFReqshift?
- :HIPmdevn
- :HIPmdevn?
- :LOAFlevel
- :LOAFlevel?
- :LOAMdepth
- :LOAMdepth?
- :LOCklevels
- :LOCklevels?
- :LOFMdevn
- :LOFMdevn?
- :LOFReqshift
- :LOFReqshift?
- :LOPmdevn
- :LOPmdevn?
- :PAuse
- :PAuse?
- :PMdecodelevel
- :PMdecodelevel?
- :READSequence?
- :READTone?
- :SENd
- :SEQuence
- :SEQuence?

:POcsagtones
- :Alert
- :Alert?
- :Bitrate
- :Bitrate?
- :Callpager
- :Fmdevn
- :Fmdevn?
- :Message
- :Message?
- :POlarity
- :POlarity?
- :PReamble
- :PReamble?
- :Ric
- :Ric?

:SEqtones
- :AFDecodelevel
- :AFDecodelevel?
- :AFLevel
- :AFLevel?
- :AMDECodelevel
- :AMDECodelevel?
- :AMDEPth
- :AMDEPth?
- :Decodereset
- :FMDECodelevel
- :FMDECodelevel?
- :FMDEVn
- :FMDEVn?
- :FReqshift
- :FReqshift?
- :Genmode
- :Genmode?
- :PMDECodelevel
- :PMDECodelevel?
- :PMDEVn
- :PMDEVn?
- :READSequence?
- :READTone?
- :SENd
- :SENd?
- :SEQuence
- :SEQuence?
- :STandard
- :STandard?

:DCstones

:AMDECodelevel

Description : Sets the ranging on the AM decode hardware so that a signal less than the set level can be decoded.

Parameters : <CPD> AMDECODE_10PC
AMDECODE_30PC
AMDECODE_100PC

Allowed suffices : Not applicable

Example : `:DCstones:AMDECodelevel AMDECODE_10PC`
Set DCS AM decode level to decode tones up to 10% modulation depth.

:AMDECodelevel?

Description : Returns the current setting of AM decode level

Response : <CRD> AMDECODE_10PC
AMDECODE_30PC
AMDECODE_100PC

Example : `:DCstones:AMDECodelevel?`

Example response : `DC:AMDEC AMDECODE_10PC`

:DCstones

:AMDEPth

Description : Sets the amplitude modulation level produced by the DCS tones signal

Parameters : <NRf>

Allowed suffices : PC

Example : `:DCstones:AMDEPth 30PC`
Set AM depth of DCS tones to 30 percent.

:AMDEPth?

Description : Returns the current setting of the DCS tones AM level

Response : <NR2> Units in %

Example : `:DCstones:AMDEPth?`

Example response : `DC:AMDEP 30PC`

:DCstones

:Bitrate

Description : Sets the bit rate at which the DCS tones signal will be generated

Parameters : <NRf>

Allowed suffices : BPS

Example : `:DCstones:Bitrate 134.4BPS`
Set DCS bit rate to 134.4 bits per second.

:Bitrate?

Description : Returns the current setting of the DCS tones bit rate

Response : <NR2> Units of bits/second

Example : `:DCstones:Bitrate?`

Example response : `DC:B 134.4BPS`

:DCstones

:Code

Description : Sets the DCS code-number to be transmitted.

Parameters : <NR1>

Allowed suffices : Not applicable

Example : `:DCstones:Code 777`
Set DCS code to 777.

:Code?

Description : Returns the current setting of the DCS tones code

Response : <NR1>

Example : `:DCstones:Code?`

Example response : `DC:C 777`

:DCstones

:FMDECodelevel

Description : Sets the ranging on the FM decode hardware so that a signal less than the set level can be decoded.

Parameters : <CPD> FMDECODE_1KHZ
FMDECODE_3KHZ
FMDECODE_10KHZ
FMDECODE_30KHZ
FMDECODE_100KHZ
FMDECODE_300KHZ

Allowed suffices : Not applicable

Example : `:DCstones:FMDECodelevel FMDECODE_1KHZ`
Set DCS FM decode level to decode tones up to 1 kHz.

:FMDECodelevel?

Description : Returns the current setting of FM decode level

Response : <CRD> FMDECODE_1KHZ
FMDECODE_3KHZ
FMDECODE_10KHZ
FMDECODE_30KHZ
FMDECODE_100KHZ
FMDECODE_300KHZ

Example : `:DCstones:FMDECodelevel?`

Example response : `DC:FMDEC FMDECODE_1KHZ`

:DCstones

:FMDEVn

Description : Sets the frequency modulation deviation level produced by the DCS tones signal.

Parameters : <NRf>

Allowed suffices : `HZ, KHZ`

Example : `:DCstones:FMDEVn 10KHZ`
Sets FM deviation of DCS tones to 10 kHz.

:FMDEVn?

Description : Returns the current setting of the DCS tones FM deviation level

Response : <NR2> Frequency in Hz

Example : `:DCstones:FMDEVn?`

Example response : `DC:FMDEV 10000Hz`

:DCstones

:Mode

Description : Sets the modulation mode, when in DCS tones screens, between DCS and normal modulation.

Parameters : <CPD> NORMAL_MOD
DCS_OFF
DCS_ON

Allowed suffices : Not applicable

Example : `:DCstones:Mode DCS_ON`
Switch on DCS tones.

:Mode?

Description : Returns the current setting of the DCS tones generator mode

Response : <CRD> NORMAL_MOD
DCS_OFF
DCS_ON

Example : `:DCstones:Mode?`

Example response : `DC:M DCS_ON`

:DCstones

:PMDECodelevel

Description : Sets the ranging on the PM decode hardware so that a signal less than the set level can be decoded.

Parameters : <CPD> PMDECODE_1RAD
PMDECODE_3RAD
PMDECODE_10RAD
PMDECODE_30RAD

Allowed suffices : Not applicable

Example : `:DCstones:PMDECodelevel PMDECODE_1RAD`
Set DCS PM decode level to decode tones up to 1 radian.

:PMDECodelevel?

Description : Returns the current setting of PM decode level

Response : <CPD> PMDECODE_1RAD
PMDECODE_3RAD
PMDECODE_10RAD
PMDECODE_30RAD

Example : `:DCstones:PMDECodelevel?`

Example response : `DC:PMDEC PMDECODE_1RAD`

:DCstones

:PMDEVn

Description : Sets the phase modulation deviation level produced by the DCS tones signal

Parameters : <NRf>

Allowed suffices : RAD, R

Example : `:DCstones:PMDEVn 10RAD`
Set PM deviation of DCS tones to 10 radians.

:PMDEVn?

Description : Returns the current setting of DCS tones PM deviation level

Response : <NR2> Units in radians

Example : `:DCstones:PMDEVn?`

Example response : `DC:PMDEV 10RAD`

:DCstones

:REadcode?

Description : Returns the decoded DCS tones with information about preferred codes. The returned string contains each code and a parameter either N or P, depending on the code being PREFERRED or NON-PREFERRED, up to a maximum of 7 codes. Query only.

Response : <ARB ASCII string>

Example : `:DCstones:REadcode?`

Example response : `DC:RE 777,P,192,N,261,N,532,N`
777 is decoded as preferred code
192 is decoded and is not preferred
261 is decoded and is not preferred
532 is decoded and is not preferred.

:DCstones

:RXpolarity

Description : Sets the polarity of the DCS decoder

Parameters : <CPD> NORMAL, INVERT

Allowed suffices : Not applicable

Example : `:DCstones:RXpolarity NORMAL`
Sets the polarity of the DCS decoder to NORMAL

:RXpolarity?

Description : Returns the current setting of the DCS tones decoder inversion function

Response : <CRD> NORMAL, INVERT

Example : `:DCstones:RXpolarity?`

Example response : `DC:RX NORMAL`

:DCstones

:Txpolarity

Description : Sets the polarity of the DCS encoder

Parameters : <CPD> NORMAL, INVERT

Allowed suffices : Not applicable

Example : `:DCstones:Txpolarity NORMAL`
Sets the polarity of the DCS encoder to NORMAL

:Txpolarity?

Description : Returns the current setting of the DCS tones generator inversion function

Response : <CRD> NORMAL, INVERT

Example : `:DCstones:Txpolarity?`

Example response : `DC:T NORMAL`

:DTmftones

:AFdecodelevel

Description : Sets the ranging on the AF decode hardware so that a signal less than the set level can be decoded (per tone).

Parameters : <CPD> AFDTMF_200MV
AFDTMF_600MV
AFDTMF_2V
AFDTMF_6V

Allowed suffices : Not applicable

Example : `:DTmftones:AFdecodelevel AFDTMF_200MV`
Set DTMF AF decode level to decode tones up to 200 mV per tone.

:AFdecodelevel?

Description : Returns the current setting of the DTMF Tones AF decode level

Response : <CRD> AFDTMF_200MV
AFDTMF_600MV
AFDTMF_2V
AFDTMF_6V

Example : `:DTmftones:AFdecodelevel?`

Example response : `DT:AF AFDTMF_200MV`

:DTmftones

:AMdecodelevel

Description : Sets the ranging on the AM decode hardware so that a signal less than the set level can be decoded (per tone)

Parameters : <CPD> AMDECODE_10PC
AMDECODE_30PC
AMDECODE_100PC

Allowed suffices : Not applicable

Example : `:DTmftones:AMdecodelevel AMDECODE_10PC`
Set DTMF AM decode level to decode tones up to 10% modulation depth per tone.

:AMdecodelevel?

Description : Returns the current setting of the DTMF Tones AM decode level

Response : <CRD> AMDECODE_10PC
AMDECODE_30PC
AMDECODE_100PC

Example : `:DTmftones:AMdecodelevel?`

Example response : `DT:AM AMDECODE_10PC`

:DTmftones

:DEcodereset

Description : Clears and resets the DTMF decoder. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:DTmftones:DEcodereset`
Reset the decoder to decode a new sequence.

:DTmftones

:DUration

Description : Sets the duration of each DTMF tone.

Parameters : <NRf>

Allowed suffices : `US, MS, S`

Example : `:DTmftones:DUration 500MS`
Set the duration of each DTMF tone to 500 ms.

:DUration?

Description : Returns the current setting of the duration of each DTMF tone.

Response : <NR2> Time in µsecs

Example : `:DTmftones:DUration?`

Example response : `DT:DU 500000US`

:DTmftones

:Fmdecodelevel

Description : Sets the ranging on the FM decode hardware so that a signal less than the set level can be decoded (per tone).

Parameters : <CPD> FMDECODE_1KHZ
FMDECODE_3KHZ
FMDECODE_10KHZ
FMDECODE_30KHZ
FMDECODE_100KHZ
FMDECODE_300KHZ

Allowed suffices : Not applicable

Example : `:DTmftones:Fmdecodelevel FMDECODE_1KHZ`
Set DTMF FM decode level to decode tones up to 1 kHz per tone.

:Fmdecodelevel?

Description : Returns the current setting of the DTMF Tones FM decode level

Response : <CRD> FMDECODE_1KHZ
FMDECODE_3KHZ
FMDECODE_10KHZ
FMDECODE_30KHZ
FMDECODE_100KHZ
FMDECODE_300KHZ

Example : `:DTmftones:Fmdecodelevel?`

Example response : `DT:F FMDECODE_1KHZ`

:DTmftones

:HIAFlevel

Description : Sets the level of AF signal produced by the high tones of the encoded DTMF.

Parameters : <NRf>

Allowed suffices : `UV, MV, V`

Example : `:DTmftones:HIAFlevel 100MV`
Sets audio signal produced by high tones to 100 mV

:HIAFlevel?

Description : Returns the current setting of AF signal produced by the high tones of the encoded DTMF

Response : <NR2> Units in µV

Example : `:DTmftones:HIAFlevel?`

Example response : `DT:HIAF 100000UV`

:DTmftones

:HIAMdepth

Description : Sets the level of amplitude modulation produced by the high tones of the encoded DTMF.

Parameters : <NRf>

Allowed suffices : PC

Example : :DTmftones:HIAMdepth 20PC
Sets the amplitude modulation level to 20%.

:HIAMdepth?

Description : Returns the current setting of the level of amplitude modulation produced by the high tones of the encoded DTMF.

Response : <NR2> Units in %

2Example : :DTmftones:HIAMdepth?

Example response : DT:HIAM 20PC

:DTmftones

:HIFMdevn

Description : Sets the level of frequency modulation deviation produced by the high tones of the encoded DTMF.

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : :DTmftones:HIFMdevn 5KHZ
Sets the deviation of the HI tone of a DTMF tone to 5 kHz.

:HIFMdevn?

Description : Returns the current setting of the level of frequency modulation deviation produced by the high tones of the encoded DTMF

Response : <NR2> Frequency in Hz

Example : :DTmftones:HIFMdevn?

Example response : DT:HIFM 5000HZ

:DTmftones

:HIFReqshift

Description : Sets the level of frequency shift applied to the high tones of the encoded sequence

Parameters : <NRf>

Allowed suffices : PC

Example : :DTmftones:HIFReqshift 2.5PC
Set a 2.5% frequency shift to all HI tones in DTMF tones.

:HIFReqshift?

Description : Returns the current setting of frequency shift applied to the high tones of the encoded sequence

Response : <NR2> Units in %

Example : :DTmftones:HIFReqshift?

Example response : DT:HIFR 2.5PC

:DTmftones

:HIPmdevn

Description : Sets the phase modulation deviation level produced by the high tones of the encoded DTMF.

Parameters : <NRf>

Allowed suffices : RAD, R

Example : :DTmftones:HIPmdevn 30 RAD
Sets the phase modulation deviation level to 30 rad.

:HIPmdevn?

Description : Returns the current setting of the phase modulation deviation level produced by the high tones of the encoded DTMF.

Response : <NR2> Units in radians

Example : :DTmftones:HIPmdevn?

Example response : DT:HIP 30RAD

:DTmftones

:LOAFlevel

Description : Sets the level of AF signal produced by the low tones of the encoded DTMF.

Parameters : <NRf>

Allowed suffices : UV, MV, V

Example : :DTmftones:LOAFlevel 100MV
Sets audio signal produced by low tones to 100 mV

:LOAFlevel?

Description : Returns the current setting of AF signal produced by the low tones of the encoded DTMF.

Response : <NR2>Units in µV

Example : :DTmftones:LOAFlevel?

Example response : DT:LOAF 100000UV

:DTmftones

:LOAMdepth

Description : Sets the level of amplitude modulation produced by the low tones of the encoded DTMF.

Parameters : <NRf>

Allowed suffices : PC

Example : :DTmftones:LOAMdepth 20PC
Sets the amplitude modulation level to 20%.

:LOAMdepth?

Description : Returns the current setting of the level of amplitude modulation produced by the low tones of the encoded DTMF.

Response : <NR2> Units in %

Example : :DTmftones:LOAMdepth?

Example response : DT:LOAM 20PC

:DTmftones

:LOCklevels

Description : Locks the levels of the high and low tones to the level set for LO TONES, and thereafter, for either setting

Parameters : <CPD> UNLOCKED, LOCKED

Allowed suffices : Not applicable

Example : `:DTmftones:LOCklevels LOCKED`
Locks the levels for high and low tones of a DTMF tone to the level set for the low tone.

:LOCklevels?

Description : Returns the current status of the locking between the DTMF tones level

Response : <CRD> UNLOCKED, LOCKED

Example : `:DTmftones:LOCklevels?`

Example response : `DT:LOC LOCKED`

:DTmftones

:LOFMdevn

Description : Sets the level of frequency modulation deviation produced by the low tones of the encoded DTMF.

Parameters : <NRf>

Allowed suffices : `HZ, KHZ`

Example : `:DTmftones:LOFMdevn 5KHZ`
Sets the deviation of the HI tone of a DTMF tone to 5 kHz.

:LOFMdevn?

Description : Returns the current setting of the level of frequency modulation deviation produced by the low tones of the encoded DTMF.

Response : <NR2> Frequency in Hz

Example : `:DTmftones:LOFMdevn?`

Example response : `DT:LOFM 5000HZ`

:DTmftones

:LOFReqshift

Description : Sets the level of frequency shift applied to the low tones of the encoded sequence

Parameters : <NRf>

Allowed suffices : PC

Example : `:DTmftones:LOFReqshift 2.5PC`
Set a 2.5% frequency shift to all LO tones in DTMF tones.

:LOFReqshift?

Description : Returns the current setting of frequency shift applied to the low tones of the encoded sequence

Response : <NR2> Units in %

Example : `:DTmftones:LOFReqshift?`

Example response : `DT:LOFR 2.5PC`

:DTmftones

:LOPmdevn

Description : Sets the phase modulation deviation level produced by the low tones of the encoded DTMF.

Parameters : <NRf>s

Allowed suffices : RAD, R

Example : `:DTmftones:LOPmdevn 30RAD`
Sets phase modulation deviation level to 30 rad.

:LOPmdevn?

Description : Returns the current setting of the phase modulation deviation level produced by the low tones of the encoded DTMF.

Response : <NR2> Units in radians

Example : `:DT:LOP?`

Example response : `DT:LOP 30RAD`

:DTmftones

:PAuse

Description : Sets the duration of the pause between each DTMF tone.

Parameters : <NRf>

Allowed suffices : US, MS, S

Example : :DTmftones:PAuse 500MS
Set the pause between each tone in a DTMF sequence to 500 ms.

:PAuse?

Description : Returns the current setting of the duration of the pause between each DTMF tone.

Response : <NR2> Time in µsecs

Example : :DTmftones:PAuse?

Example response : DT:PA 500000US

:DTmftones

:PMdecodelevel

Description : Sets the ranging on the PM decode hardware so that a signal less than the set level can be decoded (per tone)

Parameters : <CPD> PMDECODE_1RAD
PMDECODE_3RAD
PMDECODE_10RAD
PMDECODE_30RAD

Allowed suffices : Not applicable

Example : :DTmftones:PMdecodelevel PMDECODE_1RAD
Set DTMF PM decode level to decode tones up to 1 radian per tone.

:PMdecodelevel?

Description : Returns the current setting of the DTMF Tones PM decode level

Response : <CRD> PMDECODE_1RAD
PMDECODE_3RAD
PMDECODE_10RAD
PMDECODE_30RAD

Example : :DTmftones:PMdecodelevel?

Example response : DT:PM PMDECODE_1RAD

:DTmftones

:READSequence?

Description : Returns the contents of the DTMF decode register; query only

Response : <ASCII_data>

Example : `:DTmftones:READSequence?`
Read the currently decoded DTMF sequence.

Example response : `A162139ABC`

:DTmftones

:READTone?

Description : Reads the data for a particular tone from the decoded tone sequence; query only

Parameters : <NR1>

Response :

<NR1>,	Tone identity
<NR1>Hz,	Lo frequency
<NR2>%,	Lo frequency error
<NR1>Hz,	Hi frequency
<NR2>%,	Hi frequency error
<NR1>ms,	Tone duration
<NR2>dB,	Twist

Example : `:DT:READTone? 4`
Read back tone 4 information from the decoded sequence.

Example response : `2,697HZ,0.0%,1335HZ,0.1%,600MS,1.2DB`

:DTmftones

:SENd

Description : Initiates the 'Send Tones' function. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:DTmftones:SENd`
Send the DTMF sequence that is currently set up.

:DTmftones

:SEQuence

Description : Sets the character sequence to be encoded

Parameters : <string_program_data>

Allowed suffices : Not applicable

Example : `:DTmftones:SEQuence "01234567890??ABCD"`
The DTMF sequence to be sent is set to be 01234567890??ABCD

:SEQuence?

Description : Returns the current sequence set in the DTMF send register

Response : <String_response_data>

Example : `:DTmftones:SEQuence?`

Example response : `DT:SEQ 01234567890#*ABCD`

:POcsagtones

:Alert

Description : Sets the alert message type to be sent to the POCSAG device under test

Parameters : <CPD> NUMERIC_ALERT
ALERT_ONLY_TYPE_1
ALERT_ONLY_TYPE_2
TEXT_ALERT

Allowed suffices : Not applicable

Example : `:POcsagtones:Alert NUMERIC_ALERT`
Set the POCSAG alert type to NUMERIC.

:Alert?

Description : Returns the current setting of the alert message type to be sent to the POCSAG device under test

Response : <CRD> NUMERIC_ALERT
ALERT_ONLY_TYPE_1
ALERT_ONLY_TYPE_2
TEXT_ALERT

Example : `:POcsagtones:Alert?`

Example response : `PO:A NUMERIC_ALERT`

:POcsagtones

:Bitrate

Description : Sets the bit rate at which the POCSAG signal will be generated

Parameters : <NRf>

Allowed suffices : BPS

Example : `:POcsagtones:Bitrate 512BPS`
Set POCSAG bit rate to 512 Bits per sec.

:Bitrate?

Description : Returns the current setting of the bit rate at which the POCSAG signal will be generated

Response : <NR2> Units in bits/second

Example : `:POcsagtones:Bitrate?`

Example response : `PO:B 512BPS`

:POcsagtones

:Callpager

Description : Causes the generation of a POCSAG signal sequence. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:POcsagtones:Callpager`
Send the POCSAG sequence.

:POcsagtones

:Fmdevn

Description : Sets the level of frequency modulation deviation produced by the POCSAG tones signal

Parameters : <NRf>

Allowed suffices : HZ, KHZ

Example : `:POcsagtones:Fmdevn 4.5KHZ`
Set the POCSAG tones deviation to 4.5 kHz.

:Fmdevn?

Description : Returns the current setting of the level of frequency modulation deviation produced by the POCSAG tones signal

Response : <NR2> Frequency in Hz

Example : `:POcsagtones:Fmdevn?`

Example response : `PO:F 4500HZ`

:POcsagtones

:Message

Description : Sets which of the test set preset messages is to be sent to the POCSAG device under test. Setting a numeric message will automatically set the alert type to NUMERIC, and setting a text message will set the alert type to TEXT.

Parameters : <CPD> NUMERIC_MESS_1
NUMERIC_MESS_2
NUMERIC_MESS_3
TEXT_MESS_1
TEXT_MESS_2
TEXT_MESS_3
TEXT_MESS_4
TEXT_MESS_5
TEXT_MESS_6
TEXT_MESS_7
TEXT_MESS_8
TEXT_MESS_9

Allowed suffices : Not applicable

Example : `:POcsagtones:Message NUMERIC_MESS_1`
When POCSAG is switched on, send the internally stored message number 1, numeric.

:Message?

Description : Returns the current setting of the test set preset messages to be sent to the POCSAG device under test

Response : <CRD> NUMERIC_MESS_1
NUMERIC_MESS_2
NUMERIC_MESS_3
TEXT_MESS_1
TEXT_MESS_2
TEXT_MESS_3
TEXT_MESS_4
TEXT_MESS_5
TEXT_MESS_6
TEXT_MESS_7
TEXT_MESS_8
TEXT_MESS_9

Example : `:POcsagtones:Message?`

Example response : `PO:M NUMERIC_MESS_1`

:POcsagtones

:POlarity

Description : Enables or disables the POCSAG signal invert function

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:POcsagtones:POlarity ON`
Invert the POCSAG signal.

:POlarity?

Description : Returns the current status of the POCSAG signal invert function

Response : <CRD> OFF, ON

Example : `:POcsagtones:POlarity?`

Example response : `PO:PO ON`

:POcsagtones

:PReamble

Description : Enables or disables the preamble sequence of the generated POCSAG signal

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:POcsagtones:PReamble ON`
Switch on POCSAG preamble.

:PReamble?

Description : Returns the current status of preamble sequence of the generated POCSAG signal

Response : <CRD> OFF, ON

Example : `:POcsagtones:PReamble?`

Example response : `PO:PR ON`

:POcsagtones

:Ric

Description : Sets the identity code of the POCSAG device to be tested.

Parameters : <NRf>

Allowed suffices : Not applicable

Example : `:POcsagtones:Ric 1712134`
Receiver identity code is set to 1712134

:Ric?

Description : Returns the current setting of identity code of the POCSAG device to be tested.

Response : <NR1>

Example : `:POcsagtones:Ric?`

Example response : `PO:R 1712134`

:SEqtones

:AFDecodelevel

Description : Sets the ranging on the AF decode hardware so that a signal less than the set level can be decoded.

Parameters : <CPD> AFDECODE_1MV
AFDECODE_3MV
AFDECODE_10MV
AFDECODE_30MV
AFDECODE_100MV
AFDECODE_300MV
AFDECODE_1V
AFDECODE_3V
AFDECODE_10V
AFDECODE_30V

Allowed suffices : Not applicable

Example : `:SEqtones:AFDecodelevel AFDECODE_1MV`
Set Sequential Tones AF decode level to decode tones up to 1 mV.

:AFDecodelevel?

Description : Returns the current setting of Sequential Tones AF decoder level

Response : <CRD> AFDECODE_1MV
AFDECODE_3MV
AFDECODE_10MV
AFDECODE_30MV
AFDECODE_100MV
AFDECODE_300MV
AFDECODE_1V
AFDECODE_3V
AFDECODE_10V
AFDECODE_30V

Example : `:SEqtones:AFDecodelevel?`

Example response : `SE:AFD AFDECODE_1MV`

:SEqtones

:AFLevel

Description : Sets the AF level generated by sequential AF tones.

Parameters : <NRf>

Allowed suffices : UV, MV, V

Example : :SEqtones:AFLevel 100MV
Set Sequential Tones AF level to 100 mV

:AFLevel?

Description : Returns the current setting of the AF level generated by a sequential tone.

Response : <NR2> Units in µV

Example : :SEqtones:AFLevel?

Example response : SE:AFL 100000UV

:SEqtones

:AMDECodelevel

Description : Sets the ranging on the AM decode hardware so that a signal less than the set level can be decoded.

Parameters : <CPD> AMDECODE_10PC
AMDECODE_30PC
AMDECODE_100PC

Allowed suffices : Not applicable

Example : :SEqtones:AMDECodelevel AMDECODE_10PC
Set Sequential Tones AM decode level to decode tones up to 10% modulation depth.

:AMDECodelevel?

Description : Returns the current setting of Sequential Tones AM decoder level

Response : <CRD> AMDECODE_10PC
AMDECODE_30PC
AMDECODE_100PC

Example : :SEqtones:AMDECodelevel?

Example response : SE:AMDEC AMDECODE_10PC

:SEqtones

:AMDEPth

Description : Sets the amplitude modulation level produced by the Sequential Tones signal.

Parameters : <NRf>

Allowed suffices : PC

Example : `:SEqtones:AMDEPth 30PC`
Set AM depth of Sequential Tones to 30 percent.

:AMDEPth?

Description : Returns the current setting of the AM depth.

Response : <NR2> Units in %

Example : `:SEqtones:AMDEPth?`

Example response : `SE:AMDEP 30PC`

:SEqtones

:Decodereset

Description : Clears and resets the decoder. Action only.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:SEqtones:Decodereset`
Resets the decoder to decode a new sequence.

:SEqtones

:FMDECodelevel

Description : Sets the ranging on the FM decode hardware so that a signal less than the set level can be decoded.

Parameters : <CPD> FMDECODE_1KHZ
FMDECODE_3KHZ
FMDECODE_10KHZ
FMDECODE_30KHZ
FMDECODE_100KHZ
FMDECODE_300KHZ

Allowed suffices : Not applicable

Example : `:SEqtones:FMDECodelevel FMDECODE_1KHZ`
Set FM decode level to decode tones up to 1 kHz.

:FMDECodelevel?

Description : Returns the current setting of Sequential Tones FM decoder level

Response : <CRD> FMDECODE_1KHZ
FMDECODE_3KHZ
FMDECODE_10KHZ
FMDECODE_30KHZ
FMDECODE_100KHZ
FMDECODE_300KHZ

Example : `:SEqtones:FMDECodelevel?`

Example response : `SE:FMDEC FMDECODE_1KHZ`

:SEqtones

:FMDEVn

Description : Sets the deviation of FM Sequential Tones.

Parameters : <NRf>

Allowed suffices : `HZ, KHZ`

Example : `:SEqtones:FMDEVn 2.5KHZ`
Set FM Sequential Tones deviation to 2.5 kHz

:FMDEVn?

Description : Returns the current setting of the deviation of FM Sequential Tones.

Response : <NR2> Frequency in Hz

Example : `:SEqtones:FMDEVn?`

Example response : `SE:FMDEV 2500HZ`

:SEqtones

:FReqshift

Description : Sets the level of frequency shift to be applied to the Sequential Tones

Parameters : <NRf>

Allowed suffices : PC

Example : `:SEqtones:FReqshift 5PC`
Set a 5% frequency shift to all LO/HI tones in Sequential Tones.

:FReqshift?

Description : Returns the current setting of the frequency shift to be applied to the Sequential Tones.

Response : <NR2> Units in %

Example : `:SEqtones:FReqshift?`

Example response : `SE:FR 5PC`

:SEqtones

:Genmode

Description : Sets the mode of Sequential Tones generation.

Parameters : <CPD> BURST
CONT
MAN_STEP
MANUAL
NORMAL_MOD
NORMAL_AF

Allowed suffices : Not applicable

Example : `:SEqtones:Genmode BURST`
Set Sequential Tones into burst mode; only send the sequence once when told to switch on.

:Genmode?

Description : Returns the current setting of the tones generation mode

Response : <CRD> BURST
CONT
MAN_STEP
MANUAL
NORMAL_MOD
NORMAL_AF

Example : `:SEqtones:Genmode?`

Example response : `SE:G BURST`

:SEqtones

:PMDECodelevel

Description : Sets the ranging on the PM decode hardware so that a signal less than the set level can be decoded

Parameters : <CPD> PMDECODE_1RAD
PMDECODE_3RAD
PMDECODE_10RAD
PMDECODE_30RAD

Allowed suffices : Not applicable

Example : `:SEqtones:PMDECodelevel PMDECODE_1RAD`
Set PM decode level to decode tones up to 1 radian.

:PMDECodelevel?

Description : Returns the current setting of Sequential Tones PM decode level

Response : <CRD> PMDECODE_1RAD
PMDECODE_3RAD
PMDECODE_10RAD
PMDECODE_30RAD

Example : `:SEqtones:PMDECodelevel?`

Example response : `SE:PMDEC PMDECODE_1RAD`

:SEqtones

:PMDEVn

Description : Sets the deviation of PM Sequential Tones

Parameters : <NRf>

Allowed suffices : `RAD, R`

Example : `:SEqtones:PMDEVn 15RAD`
Set PM Sequential Tones deviation to 15 rad

:PMDEVn?

Description : Returns the deviation of PM Sequential Tones.

Response : <NR2> Units in radians

Example : `:SEqtones:PMDEVn?`

Example response : `SE:PMDEV 15RAD`

:SEqtones

:READSequence?

Description : Returns the decoded sequence

Response : <ASCII_data>

Example : `SEqtones:READSequence?`
Read the currently decoded Sequential Tones sequence

Example response : `1234567890ABCDE`

:SEqtones

:READTone?

Description : Reads a particular tone from the decoded tone sequence.

Response : <ASCII_data>

Example : `:SEqtones:READTone? 12`
Read tone 12 from the decoded sequence

Example response : `B`

:SEqtones

:SENd

Description : Enables or disables the 'Send Tones' function

Parameters : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:SEqtones:SENd ON`
Turn on the Send Tones function.

:SENd?

Description : Returns the current status of Sequential Tones 'Send Tones' function

Response : <CRD> OFF, ON

Example : `:SEqtones:SENd?`

Example response : `SE:SEN ON`

:SEqtones

:SEQuence

Description : Sets the character sequence to be encoded

Parameters : <String_program_data>

Allowed suffices : Not applicable

Example : `:SEqtones:SEQuence "1234567890ABCDE"`
Set up the sequence to be 1234567890ABCDE

:SEQuence?

Description : Returns the current sequence set in the Sequential Tones generator register

Response : <String_response_data>

Example : `:SEqtones:SEQuence?`

Example response : `SE:SEQ 1234567890ABCDE`

:SEqtones

:STandard

Description : Sets the Sequential Tones code standard.

Parameters : <CPD> CCIR1
CCIR2
CCIR3
CCIR4
CCIR5
ZVEI1
ZVEI2
PZVEI
ZVEI3
DZVEI
EEA
EIA
USER1
USER2
USER3

Allowed suffices : Not applicable

Example : `:SEqtones:STandard CCIR1`
Set Sequential Tones standard to CCIRI.

:STandard?

Description : Returns the current setting of Sequential Tones code standard

Response : <CRD> CCIR1
CCIR2
CCIR3
CCIR4
CCIR5
ZVEI1
ZVEI2
PZVEI
ZVEI3
DZVEI
EEA
EIA
USER1
USER2
USER3

Example : `:SEqtones:STandard?`

Example response : `SE:ST CCIR1`

Chapter 6
SYSTEMS COMMANDS

Contents

Introduction 6-1
Command abbreviations 6-1
Program download and run commands (:PRog) 6-2
List of commands 6-2
Test commands (:STEst) 6-7
List of commands 6-7
Test parameter commands (:PAram) 6-27
List of commands 6-27
Control commands (:SYCtrl) 6-169
List of commands 6-169
Manual commands (:SYManual) 6-170
Introduction 6-170
List of commands 6-170
Setup commands (:SYSetup) 6-176
List of commands 6-176
Setup commands for user-defined systems (:SYUsersetup) 6-226
List of commands 6-226

Introduction

This chapter describes System commands, that is, those commands that are specifically intended for use in System test programs.

Command abbreviations

The command syntax provided in this chapter, as in Chapter 5, indicates the minimum number of characters that must be entered for a command to be valid by showing them in uppercase. For example, the command :PAram can be entered as just :PA. Conversely, the complete command, :PARAM, is equally valid, and more readily understandable. Another consideration is that the addition of new commands in future development could invalidate the currently acceptable minimum number of characters. It is therefore recommended that, where possible, the complete command is used.

Program download and run commands (:PRog)

List of commands

:PRog
- **:Clrstore**
- **:LEARN**
- **:LEARN?**
- **:LEARNLine**
- **:Tsi or :SEquence**
- **:Tsi? or :SEquence?**
- **:PAuse**
- **:PAuse?**
- **:PRintstore**
- **:Runstate**
- **:Runstate?**
- **:Numresults?**
- **:STrresults?**

:PRog

:Clrstore

Description : Clears the currently active Autorun test results window.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:PRog:Clrstore`

:PRog

:LEARN

Description : Set the test set into state where an MI-BASIC program can be downloaded.

Parameter : <CPD> ACTIVE
INACTIVE

Allowed suffices : Not applicable

Example : `:PRog:LEARN ACTIVE`

:LEARN?

Description : Returns the MI-BASIC program download status of the test set.

Response : <CRD> ACTIVE
INACTIVE

Example : `:PRog:LEARN?`

Example response : `PR:LEARN ACTIVE`

:PRog

:LEARNLine

Description : Downloads the next line of an MI-BASIC program.

Parameters : <ASCII data>

Allowed suffices : Not applicable

Example :
```
:PRog:LEARNLine 'LPRINT "1.  Program 1"'
```
(See Part 1, Chapter 1, for more examples.)

:PRog

:Tsi or :SEquence

Description : Selects the test program to be run.

Parameter : <CPD> CALL_PROC
CALL_RF
BRIEF
COMPREHENSIVE
USER
GEN_PMR

Allowed suffices : Not applicable

Examples :
```
:PRog:Tsi CALL_PROC
:PRog:SEquence CALL_PROC
```

:Tsi? or :SEquence?

Description : Returns the currently selected test program.

Response : <CRD> CALL_PROC
CALL_RF
BRIEF
COMPREHENSIVE
USER
GEN_PMR

Example :
```
:PRog:Tsi?
:PRog:SEquence?
```

Example response :
```
PR:T CALL_PROC
PR:SE CALL_PROC
```

:PRog

:PAuse

Description : Sets the test set's Autorun test pause state.

Parameter : <CPD> OFF
ON_FAILURE
ALWAYS

Allowed suffices : Not applicable

Example : `:PRog:PAuse OFF`

:PAuse?

Description : Returns the current setting of the test set's Autorun test pause state.

Response : <CPR> OFF
ON_FAILURE
ALWAYS

Example : `:PRog:PAuse?`

Example response : `PR:PA OFF`

:PRog

:PRintstore

Description : Prints the contents of the currently active Autorun test results window to the printer.

Parameters : Not applicable

Allowed suffices : Not applicable

Example : `:PRog:PRintstore`

:PRog

:Runstate

Description : Controls the running of the test set's Autorun test program.

Parameter : <CPD> STOP
RUN
PAUSE
CONTINUE

Allowed suffices : Not applicable

Example : `:PRog:Runstate RUN`
Run the program

: Runstate?

Description : Returns the current setting of the Autorun test program control.

Response : <CRD> STOP
RUN
PAUSE
CONTINUE

Example : `:PRog:Runstate?`

Example response : `PR:R RUN`

:PRog

:Numresults?

Description : Returns the numerical test results. For further information, see the description of the command NUMRESULTS in Chapter 2.

Response : <CRD> A single ASCII string with each of the 8 numerical results, comma-separated:-

Data field 1, PASSED
Data field 2, ACTUAL
Data field 3, TARGET
Data field 4, LOWER
Data field 5, UPPER
Data field 6, AUX1
Data field 7, AUX2
Data field 8, AUX3

Example : `:PRog:Numresults?`

Example response : `PR:N 0,38.636,0.000,25.000,0.000,0.000,0.000,0.000`
Response to RxSINAD test where the test passed at 38.6 dB with a minimum pass value of 25 dB

:PRog

:STrresults?

Description : Returns the string test results. For further information, see the description of the command STRRESULTS in Chapter 2.

Response : <CRD> A single ASCII string with each of the 8 string results, comma-separated:-

Data field 1, TITLE
Data field 2, STATUS
Data field 3, COMMENT1
Data field 4, COMMENT2
Data field 5, AUX1
Data field 6, AUX2
Data field 7, AUX3
Data field 8, AUX4

Example : `:PRog:STrresults?`

Example response :
```
PR:ST REGISTRATION,PASSED,MIN:000-0-000000,,
ESN:15/20/00/26346STD1,CLASS:4,DTX:0,
CHANNELS:1320
```
Response to a TACS registration test

Test commands (:STEst)

The following :STEst commands allow you to run the System tests remotely. When the command is executed, the screen changes (if necessary) to the AUTORUN screen and the test executes in the normal way. Details of the tests that are used in the built-in System test programs can be found in the Operating Manual Supplement for the particular system in use. The test parameters can be altered remotely by using the appropriate :PAram command before running the test. These commands are detailed in the *Test parameter commands* section of this chapter (see page 6-27).

If the test passes, a value of '1' is returned. Otherwise the test fails and a value of '0' is returned.

To get the test results, use the :PRog:Numresults? or :PRog:STrresults? command after the test has completed. The complete list of :PRog commands is given earlier in this chapter (see page 6-2).

List of commands

:STEst

- :AFFreq?
- :AFLevel?
- :DATADevn?
- :DATAPerform?
- :DHandoff?
- :DSatdevn?
- :DTmfdecode?
- :Endtch?
- :Fmdevn?
- :HAndoff?
- :HOokflash?
- :LAndclear?
- :LOopbk?
- :MOBileclear?
- :MOBLev?
- :MODfreq?
- :PAgemobile?
- :PKPhase?
- :PLacecall?
- :POwerlevel?
- :PRofile?
- :PTTOFf?
- :PTTON?
- :REGIster?
- :REGWarp?
- :RMSPhase?
- :RXBERI?
- :RXBERII?
- :RXDistn?
- :RXExpand?
- :RXFer?
- :RXRBERIB?
- :RXRBERII?
- :RXRSsi?
- :RXSEns?
- :RXSInad?
- :RXSN?
- :SATDevn?
- :SATFreq?
- :SPeechqual?
- :STArttch?
- :STDEvn?
- :STDUrn?
- :STFreq?
- :TIMAdv?
- :TIMIng?
- :TXCompress?
- :TXDistn?
- :TXFreq?
- :TXLEvel?
- :TXLImit?
- :TXModsens?
- :TXNoise?
- :TXSInad?
- :TXSN?

:STEst

Applicable to AMPS, NAMPS, MPT1327, NMT, PMR, TACS and NTACS

:AFFreq?

Description : Runs the Audio Frequency measurement test. The test passes if the result is within the allowed % error.
Note: This test is not included in the built-in System test programs but is available for use in user-defined programs.

Response : <NR1> 0 or 1

Example : `:STEst:AFFreq?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, MPT1327, NMT, PMR, TACS and NTACS

:AFLevel?

Description : Runs the AF Level test. The test passes if the result is within the allowed % error.
Note: This test is not included in the built-in System test programs but is available for use in user-defined programs.

Response : <NR1> 0 or 1

Example : `:STEst:AFLevel?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS and TACS

:DATADevn?

Description : Runs the Data Deviation test. The test passes if the result is within the allowed % error.

Response : <NR1> 0 or 1

Example : `:STEst:DATADevn?`

Example response : `1`

Channel type : Wide-band Voice only

:STEst

Applicable to AMPS and TACS

:DATAPerform?

Description : Runs the Data Performance test. The test passes if the result is better than the allowed % error.

Response : <NR1> 0 or 1

Example : `:STEst:DATAPerform?`

Example response : `1`

Channel type : Wide-band Voice only

:STEst

Applicable to GSM, DCS and PCS

:DHandoff?

Description : Runs the Digital Handoff test. The test passes if a handoff to the specified channel and slot is completed within the timeout period. If the test set is in TCH Test mode, this command will change the channel without going through any channel assignment protocol (see STEST:STArttch on page 6-21).
See also the :SYSetup commands :FTCS (page 6-190), :HANDOFFlS (page 6-192), and :LTCS (page 6-198).

Parameters : <NR1> Channel number
<NR1> Slot number

Response : <NR1> 0 or 1

Example : `:STEst:DHandoff? 100,7`

Example response : `1`

Channel type : Traffic only

:STEst

Applicable to NAMPS and NTACS

:DSatdevn?

Description : Runs the Digital SAT Deviation test. The test passes if the result is within the allowed % error.

Response : <NR1> 0 or 1

Example : `:STEst:DSatdevn?`

Example response : `1`

Channel type : Narrow-band Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS and PMR

:DTmfdecode?

Description : Runs the DTMF Decode test. The test passes if the required tone sequence is received correctly within the timeout period.

Response : <NR1> 0 or 1

Example : `:STEst:DTmfdecode?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to GSM, DCS and PCS

:Endtch?

Description : Returns the test set from TCH Test mode to generating a control channel. The test passes if the test set ends TCH Test mode.
See also :STArttch?, page 6-21.
Note: This test is not included in the built-in System test programs but is available for use in user-defined programs. Further details are given in the GSM/DCS/PCS Supplement.

Response : <NR1> 0 or 1

Example : `:STEst:Endtch?`

Example response : 1

Channel type : Traffic only (TCH Test mode)

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:Fmdevn?

Description : Runs the FM Deviation test. The test passes if the result is within the allowed % error.
Note: This test is not included in the built-in System test programs but is available for use in user-defined programs.

Response : <NR1> 0 or 1

Example : `:STEst:Fmdevn?`

Example response : 1

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327 and NMT

:HAndoff?

Description : Runs the Handoff test. The test passes if a handoff to the specified channel is completed within the timeout period.

Parameter : <NR1> Channel number

Response : <NR1> 0 or 1

Example : `:STEst:HAndoff? 100`

Example response : 1

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS and NTACS

:HOokflash?

Description : Runs the Hookflash test. The test passes if a hookflash message is received within the timeout period.

Response : <NR1> 0 or 1

Example : `:STEst:HOokflash?`

Example response : 1

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:LAndclear?

Description : Runs the Clear From Land test. The test passes if the mobile is cleared down within the timeout period.

Response : <NR1> 0 or 1

Example : `:STEst:LAndclear?`

Example response : 1

Channel type : Traffic or Voice only

:STEst

Applicable to GSM, DCS and PCS

:LOopbk?

Description : Runs the Loopback test with the specified mode. The test passes if the mobile enters, or is already in, the specified mode.
Note: This test is not included in the built-in System test programs but is available for use in user-defined programs. Further details are given in the GSM/DCS/PCS Supplement.

Parameter : <CPD> RBER_LOOPBACK
BER_LOOPBACK
NO_LOOPBACK

Allowed suffices : Not applicable

Response : <NR1> 0 or 1

Example : `:STEst:LOopbk? RBER_LOOPBACK`

Example response : 1

Channel type : Traffic only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:MOBIleclear?

Description : Runs the Clear From Mobile test. The test passes if the mobile clears down within the timeout period.

Response : <NR1> 0 or 1

Example : `:STEst:MOBIleclear?`

Example response : 1

Channel type : Traffic or Voice only

:STEst

Applicable to GSM, DCS and PCS

:MOBLev?

Description : Runs the Mobile Power Level test at the specified power level. The test passes if the mobile's reported power level is the same as the specified power level.
Note: This test is not included in the built-in System test programs but is available for use in user-defined programs. Further details are given in the GSM/DCS/PCS Supplement.

Parameter : <NR1> Power level, range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS
If the specified level is invalid for the class of mobile, the test is run using the nearest valid level.

Allowed suffices : Not applicable

Response : <NR1> 0 or 1

Example : `:STEst:MOBLev? 12`

Example response : 1

Channel type : Traffic only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:MODfreq?

Description : Runs the Modulation Frequency test. The test passes if the result is within the allowed % error.
Note: This test is not included in the built-in System test programs but is available for use in user-defined programs.

Response : <NR1> 0 or 1

Example : `:STEst:MODfreq?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PAgemobile?

Description : Runs the Page Mobile test. The test passes if the mobile goes into "conversation" within the timeout period.

Response : <NR1> 0 or 1

Example : `:STEst:PAgemobile?`

Example response : `1`

Channel type : Control or Combined CC/TC (NMT only) or Calling (NMT only).

:STEst

Applicable to GSM, DCS and PCS

:PKPhase?

Description : Runs the Tx Peak Phase Error test. The test passes if the result is below the upper limit.

Response : <NR1> 0 or 1

Example : `:STEst:PKPhase?`

Example response : `1`

Channel type : Traffic only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PLacecall?

Description : Runs the Place Call test. The test passes if the mobile goes into "conversation" within the timeout period.

Response : <NR1> 0 or 1

Example : `:STEst:PLacecall?`

Example response : `1`

Channel type : Control or Traffic or Combined CC/TC (NMT only) or Access (NMT only).

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, GSM, DCS and PCS

:POwerlevel?

Description : Runs the Power Level test at the specified level. The test passes if the result is within the allowed error.

Parameter : <NR1> Range dependent on system selected:-
0-7 for AMPS and TACS
0-3 for NMT
0-19 for GSM
29-31 and 0-15 for DCS
30-31 and 0-15 for PCS
For GSM, DCS and PCS, if the specified level is invalid for the mobile Class, the test is run at the nearest valid level.

Response : <NR1> 0 or 1

Example : `:STEst:POwerlevel? 3`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to GSM, DCS and PCS

:PRofile?

Description : Runs the Power Profile test at the specified level. The test passes if the result is within GSM/DCS/PCS defined limits.

Parameter : <NR1> 0-19 for GSM
29-31 and 0-15 for DCS
30-31 and 0-15 for PCS

Response : <NR1> 0 or 1

Example : `:STEst:PRofile? 4`

Example response : `1`

Channel type : Traffic only

:STEst

Applicable to MPT1327 and PMR

:PTTOFf?

Description : Tests that the mobile's Press To Talk (PTT) button is in the released (OFF) state. The test passes if the mobile stops transmitting within the timeout period.

Parameter : <NR1> 0 or 1

Example : `:STEst:PTTOFf?`

Example response : `1`

Channel Type : Traffic or Voice only

:STEst

Applicable to MPT1327 and PMR

:PTTON?

Description : Tests that the mobile's Press To Talk (PTT) button is in the pressed (ON) state. The test passes if the mobile starts transmitting within the timeout period.

Response : <NR1> 0 or 1

Example : `:STEst:PTTON?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:REGIster?

Description : Tests that the mobile can register. The test passes if the mobile registers within the timeout period.

Response : <NR1> 0 or 1

Example : `:STEst:REGIster?`

Example response : `1`

Channel type : Control or Combined CC/TC (NMT only).

:STEst

Applicable to GSM, DCS and PCS

:REGWarp?

Description : Tests the ability of the mobile to register, with the instrument's carrier frequency and data rate offset as specified. The test passes if the mobile registers within the timeout period.

Parameter : <CPD> OFF (no offset)
HIGH (+25 parts/10^6 offset)
LOW (–25 parts/10^6 offset)

Response : <NR1> 0 or 1

Example : `:STEst:REGWarp? HIGH`

Example response : 1

Channel type : Control only

:STEst

Applicable to GSM, DCS and PCS

:RMSPhase?

Description : Runs the Tx RMS Phase Error test. The test passes if the result is below the upper limit.

Response : <NR1> 0 or 1

Example : `:STEst:RMSPhase?`

Example response : 1

Channel type : Traffic only

:STEst

Applicable to GSM, DCS and PCS

:RXBERI?

Description : Runs the Class I Bit Error Rate test. The test passes if the result is below the upper % error.

Response : <NR1> 0 or 1

Example : `:STEst:RXBERI?`

Example response : 1

Channel type : Traffic only

:STEst

Applicable to GSM, DCS and PCS

:RXBERII?

Description : Runs the Class II Bit Error Rate test. The test passes if the result is below the upper % error.

Response : <NR1> 0 or 1

Example : `:STEst:RXBERII`

Example response : `1`

Channel type : Traffic only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:RXDistn?

Description : Runs the Receiver Distortion test. The test passes if the result is better than the allowed % limit.

Response : <NR1> 0 or 1

Example : `:STEst:RXDistn?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:RXExpand?

Description : Runs the Receiver Expansion test. The test passes if the result is within the allowed % error.

Response : <NR1> 0 or 1

Example : `:STEst:RXExpand?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to GSM, DCS and PCS

:RXFer?

Description : Runs the Receiver Frame Erasure Rate test. The test passes if the result is below the upper % error.

Response : <NR1> 0 or 1

Example : `:STEst:RXFer?`

Example response : `1`

Channel type : Traffic only

:STEst

Applicable to GSM, DCS and PCS

:RXRBERIB?

Description : Runs the Class IB Residual Bit Error Rate test. The test passes if the result is below the upper % error.

Response : <NR1> 0 or 1

Example : `:STEst:RXRBERIB?`

Example response : `1`

Channel type : Traffic only

:STEst

Applicable to GSM, DCS and PCS

:RXRBERII?

Description : Runs the Class II Residual Bit Error Rate test. The test passes if the result is below the upper % error.

Response : <NR1> 0 or 1

Example : `:STEst:RXRBERII?`

Example response : `1`

Channel type : Traffic only

:STEst

Applicable to GSM, DCS and PCS

:RXRSsi?

Description : Runs the Received Signal Strength Indication test. The test passes if the result is below the upper dB limit at the specified RF generator level.

Parameter : <NRf> RF generator level, range –135 to –15

Allowed suffices : DBM

Response : <NR1> 0 or 1

Example : `:STEst:RXRSsi? -48.5DBM`

Example response : `1`

Channel type : Traffic only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:RXSEns?

Description : Runs the Receiver Sensitivity test. The test passes if the result is better than the RF upper dBm limit.

Response : <NR1> 0 or 1

Example : `:STEst:RXSEns?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:RXSInad?

Description : Runs the Receiver SINAD test. The test passes if the result is better than the lower dB limit.

Response : <NR1> 0 or 1

Example : `:STEst:RXSInad?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:RXSN?

Description : Runs the Receiver S/N test. The test passes if the result is better than the lower dB limit.

Response : <NR1> 0 or 1

Example : `:STEst:RXSN?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, TACS and NMT

:SATDevn?

Description : Runs the SAT Deviation test. The test passes if the result is within the allowed % error.

Response : <NR1> 0 or 1

Example : `:STEst:SATDevn?`

Example response : `1`

Channel type : Wide-band Voice or Traffic only

:STEst

Applicable to AMPS, TACS and NMT

:SATFreq?

Description : Runs the SAT Frequency test. The test passes if the result is within the allowed % error.

Response : <NR1> 0 or 1

Example : `:STEst:SATFreq?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to GSM, DCS and PCS

:SPeechqual?

Description : Runs the Speech Quality test. The test passes if the user considers the speech quality to be good enough.

Response : <NR1> 0 or 1

Example : `:STEst:SPeechqual?`

Example response : `1`

Channel type : Traffic only

:STEst

Applicable to GSM, DCS and PCS

:STArttch?

Description : Puts the test set into TCH Test mode on the specified channel. The test passes if the test set enters TCH Test mode.
If the specified channel is not available, the next available higher channel is used.
Note: This test is not included in the built-in System test programs but is available for use in user-defined programs. Further details are given in the GSM/DCS/PCS Supplement. See also :Endtch?, page 6-10.

Parameter : <NR1> Channel number

Allowed suffices : Not applicable

Response : <NR1> 0 or 1

Example : `:STEst:STArttch? 100`

Example response : `1`

Channel type : Control only

:STEst

Applicable to AMPS and TACS

:STDEvn?

Description : Runs the ST Deviation test. The test passes if the result is within the allowed % error. This test requires that the :PAram:STDEvn:Status ON command has been run, followed by :STEst:PAgemobile. The Page Mobile test measures the ST deviation, which is then returned when this test is run.

Response : <NR1> 0 or 1

Example : `:STEst:STDEvn?`

Example response : `1`

Channel type : Wide-band Voice only

:STEst

Applicable to AMPS and TACS

:STDUrn?

Description : Runs the ST Duration test. The test is performed against the most recent valid ST signalling context. For example, if the ST Duration test is requested after a hookflash on TACS, then the test will check that the ST was present for 400 ms.

The test passes if the result is within the allowed % error.

Response : <NR1> 0 or 1

Example : `:STEst:STDUrn?`

Example response : `1`

Channel type : Wide-band Voice only

:STEst

Applicable to AMPS and TACS

:STFreq?

Description : Runs the ST Frequency test. The test passes if the result is within the allowed % error. This test requires that the :PAram:STFreq:Status ON command has been run, followed by :STEst:PAgemobile. The Page Mobile test measures the ST frequency, which is then returned when this test is run.

Response : <NR1> 0 or 1

Example : `:STEst:STFreq?`

Example response : `1`

Channel type : Wide-band Voice only

:STEst

Applicable to GSM, DCS and PCS

:TIMAdv?

Description : Runs the Timing Advance test at the specified timing advance. The test passes if the mobile's reported timing advance is the same as the specified timing advance.
Note: This test is not included in the built-in System test programs but is available for use in user-defined programs. Further details are given in the GSM/DCS/PCS Supplement.

Parameter : <NRf> Timing advance, range 0 to 63 bits

Allowed suffices : BITS

Response : <NR1> 0 or 1

Example : `:STEst:TIMAdv? 20BITS`

Example response : 1

Channel type : Traffic only

:STEst

Applicable to GSM, DCS and PCS

:TIMIng?

Description : Runs the Timing test at the specified timing advance. The test passes if the result is within the error limit.

Parameter : <NRf> Timing advance, range 0 to 63 bits

Allowed suffices : BITS

Response : <NR1> 0 or 1

Example : `:STEst:TIMIng? 20BITS`

Example response : 1

Channel type : Traffic only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:TXCompress?

Description : Runs the Transmitter Compression test. The test passes if the result is within the allowed % error.

Response : <NR1> 0 or 1

Example : `:STEst:TXCompress?`

Example response : -

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:TXDistn?

Description : Runs the Transmitter Distortion test. The test passes if the result is better than the upper % limit.

Response : <NR1> 0 or 1

Example : `:STEst:TXDistn?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:TXFreq?

Description : Runs the Transmitter Frequency test. The test passes if the result is within the allowed % error.

Response : <NR1> 0 or 1

Example : `:STEst:TXFreq?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:TXLEvel?

Description : Runs the Transmitter Level test. The test passes if the result is within the allowed dBm limits.

Response : <NR1> 0 or 1

Example : `:STEst:TXLEvel?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:TXLImit?

Description : Runs the Transmitter Limiting test. The test passes if the result is better than the upper limit.

Response : <NR1> 0 or 1

Example : `:STEst:TXLImit?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:TXModsens?

Description : Runs the Transmitter Modulation Sensitivity test. The test passes if the result is within the allowed levels.

Response : <NR1> 0 or 1

Example : `:STEst:TXModsens?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:TXNoise?

Description : Runs the Transmitter Noise test. The test passes if the result is better than the upper noise limit.

Response : <NR1> 0 or 1

Example : `:STEst:TXNoise?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:TXSInad?

Description : Runs the Transmitter SINAD test. The test passes if the result is better than the lower dB limit.

Response : <NR1> 0 or 1

Example : `:STEst:TXSInad?`

Example response : `1`

Channel type : Traffic or Voice only

:STEst

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT and PMR

:TXSN?

Description : Runs the Transmitter S/N test. The test passes if the result is better than the lower dB limit.

Response : <NR1> 0 or 1

Example : `:STEst:TXSN?`

Example response : `1`

Channel type : Traffic or Voice only

Test parameter commands (:PAram)

List of commands

:PAram
- :AFFreq
  - :Error
  - :Error?
  - :Ref
  - :Ref?
- :AFLevel
  - :Error
  - :Error?
  - :Ref
  - :Ref?
- :Configuration
  - :Duplex
  - :Duplex?
  - :Mblevel
  - :Mblevel?
  - :Rfgenlevel
  - :Rfgenlevel?
  - :Timeadvance
  - :Timeadvance?
  - :TXsetting
  - :TXsetting?
- :DATADevn
  - :Error
  - :Error?
  - :Ref
  - :Ref?
  - :Status
  - :Status?
- :DATAPerform
  - :Lower
  - :Lower?
  - :Rfgenlevel
  - :Rfgenlevel?
  - :Status
  - :Status?
- :DHandoff
  - :Rfgenlevel
  - :Rfgenlevel?
  - :Status
  - :Status?
- :DSatdevn
  - :Averages
  - :Averages?
  - :Error
  - :Error?
  - :Ref
  - :Ref?
  - :Status
  - :Status?
- :DTmfdecode
  - :Accport
  - :Accport?
  - :Status
  - :Status?
  - :Timeout
  - :Timeout?
  - :Useaccport
  - :Useaccport?
- :Fmdevn
  - :Error
  - :Error?
  - :Ref
  - :Ref?
- :HAndoff
  - :Chantype
  - :Chantype?
  - :Mblevel
  - :Mblevel?
  - :Rfgenlevel
  - :Rfgenlevel?
  - :Timeout
  - :Timeout?
- :HOokflash
  - :Accport
  - :Accport?
  - :Status
  - :Status?
  - :Timeout
  - :Timeout?
  - :Useaccport
  - :Useaccport?
- :Landclear
  - :Rfgenlevel
  - :Rfgenlevel?
  - :Status
  - :Status?
- :MOBileclear
  - :Accport
  - :Accport?
  - :Rfgenlevel
  - :Rfgenlevel?
  - :Status
  - :Status?
  - :Timeout
  - :Timeout?
  - :Useaccport
  - :Useaccport?
- :MODfreq
  - :Error
  - :Error?
  - :Ref
  - :Ref?
- :PAgemobile
  - :Accport
  - :Accport?
  - :Chantype
  - :Chantype?
  - :Mblevel
  - :Mblevel?
  - :Rfgenlevel
  - :Rfgenlevel?
  - :Status
  - :Status?
  - :Timeout
  - :Timeout?
  - :Useaccport
  - :Useaccport?
- :PKPhase
  - :Bursttst
  - :Bursttst?
  - :Meastype
  - :Meastype?
  - :Status
  - :Status?
  - :Upper
  - :Upper?
- :PLacecall
  - :Accport
  - :Accport?
  - :Chantype
  - :Chantype?
  - :Mblevel
  - :Mblevel?
  - :Rfgenlevel
  - :Rfgenlevel?
  - :Status
  - :Status?
  - :Timeout
  - :Timeout?
  - :Useaccport
  - :Useaccport?

:POwerlevel
:Averages
:Averages?
:Firstvalue
:Firstvalue?
:HIRLower
:HIRLower?
:HIRUpper
:HIRUpper?
:Increment
:Increment?
:LAstvalue
:LAstvalue?
:LORLower
:LORLower?
:LORUpper
:LORUpper?
:MIDRLower
:MIDRLower?
:MIDRUpper
:MIDRUpper?
:RLower
:RLower?
:RUpper
:RUpper?
:Status
:Status?
:PRofile
:Firstvalue
:Firstvalue?
:Increment
:Increment?
:Lastvalue
:Lastvalue?
:Status
:Status?
:PTTOFf
:Lref
:Lref?
:Timeout
:Timeout?
:Usepressel
:Usepressel?
:PTTON
:Accport
:Accport?
:Lref
:Lref?
:Timeout
:Timeout?
:USEAccport
:USEAccport?
:USEPressel
:USEPressel?
:REGIster
:ACcport
:ACcport?
:AUthchk
:AUthchk?
:Mblevel
:Mblevel?
:REtryenabled
:REtryenabled?
:RFgenlevel
:RFgenlevel?
:Status
:Status?
:Timeout
:Timeout?
:Useaccport
:Useaccport?
:REGWarp
:AUthchk
:AUthchk?
:REtryenabled
:REtryenabled?
:RFgenlevel
:RFgenlevel?
:Status
:Status?
:Timeout
:Timeout?
:Warptype
:Warptype?
:RMSPhase
:Bursttst
:Bursttst?
:Meastype
:Meastype?
:Status
:Status?
:Upper
:Upper?
:RXBERI
:Hirfgenlevel
:Hirfgenlevel?
:Rfgenlevel
:Rfgenlevel?
:SAmples
:SAmples?
:STatus
:STatus?
:Upper
:Upper?
:RXBERII
:Hirfgenlevel
:Hirfgenlevel?
:Rfgenlevel
:Rfgenlevel?
:SAmples
:SAmples?
:STatus
:STatus?
:Upper
:Upper?
:RXDistn
:Averages
:Averages?
:Modlevel
:Modlevel?
:Nbmodlevel
:Nbmodlevel?
:RFgenlevel
:RFgenlevel?
:RXfilter
:RXfilter?
:Status
:Status?
:Upper
:Upper?
:RXExpand
:Averages
:Averages?
:Error
:Error?
:Modlevel
:Modlevel?
:Nbmodlevel
:Nbmodlevel?
:REf
:REf?
:RXfilter
:RXfilter?
:STatus
:STatus?
:SWing
:SWing?
:RXFer
:Hirgfenlevel
:Hirfgenlevel?
:RFgenlevel
:Rfgenlevel?
:SAmples
:SAmples?
:STatus
:STatus?
:Upper
:Upper?
:RXRBERIB
:Hirfgenlevel
:Hirfgenlevel?
:Rfgenlevel
:Rfgenlevel?
:SAmples
:SAmples?
:STatus
:STatus?

:Upper
:Upper?
:RXRBERII
:Hirfgenlevel
:Hirfgenlevel?
:Rfgenlevel
:Rfgenlevel?
:SAmples
:SAmples?
:STatus
:STatus?
:Upper
:Upper?
:RXRSsi
:Decrement
:Decrement?
:Error
:Error?
:Firstvalue
:Firstvalue?
:Hierror
:Hierror?
:Lastvalue
:Lastvalue?
:Status
:Status?
:RXSEns
:Averages
:Averages?
:Hiupper
:Hiupper?
:Modlevel
:Modlevel?
:Nbmodlevel
:Nbmodlevel?
:REFBer
:REFBer?
:REFSinad
:REFSinad?
:RXfilter
:RXfilter?
:SAmples
:SAmples?
:STatus
:STatus?
:Upper
:Upper?
:RXSInad
:Averages
:Averages?
:Lower
:Lower?
:Modlevel
:Modlevel?
:Nbmodlevel
:Nbmodlevel?
:RFgenlevel
:RFgenlevel?
:RXfilter
:RXfilter?
:Status
:Status?
:RXSN
:Averages
:Averages?
:Lower
:Lower?
:Modlevel
:Modlevel?
:Nbmodlevel
:Nbmodlevel?
:RFgenlevel
:RFgenlevel?
:RXfilter
:RXfilter?
:Status
:Status?
:SATDevn
:Error
:Error?
:Ref
:Ref?
:Status
:Status?
:SATFreq
:Error
:Error?
:Ref
:Ref?
:Status
:Status?
:SPEechqual
:RFgenlevel
:RFgenlevel?
:Status
:Status?
:Timeout
:Timeout?
:SPOtfreq
:Rxchann *n*,
:Rxchann? *n*
:Status
:Status?
:Txchann *n*,
:Txchann? *n*
:STDEvn
:Error
:Error?
:Ref
:Ref?
:Status
:Status?
:STDUrn
:Error
:Error?
:Ref
:Ref?
:Status
:Status?
:STFreq
:Error
:Error?
:Ref
:Ref?
:Status
:Status?
:TIming
:Error
:Error?
:Firstvalue
:Firstvalue?
:Increment
:Increment?
:Lastvalue
:Lastvalue?
:Status
:Status?
:TXCompress
:Averages
:Averages?
:Error
:Error?
:Modlevel
:Modlevel?
:Nbmodlevel
:Nbmodlevel?
:Ref
:Ref?
:STatus
:STatus?
:SWing
:SWing?
:Txfilter
:Txfilter?
:TXDistn
:Averages
:Averages?
:Modlevel
:Modlevel?
:Nbmodlevel
:Nbmodlevel?
:Status
:Status?
:Txfilter
:Txfilter?
:Upper
:Upper?

:TXFreq
:Bursttst
:Bursttst?
:Error
:Error?
:Meastype
:Meastype?
:Nberror
:Nberror?
:Ref
:Ref?
:Status
:Status?
:TXLEvel
:Averages
:Averages?
:Lower
:Lower?
:Status
:Status?
:Upper
:Upper?
:TXLImit
:Averages
:Averages?
:Modlevel
:Modlevel?
:NBModlevel
:NBModlevel?
:NBUpper
:NBUpper?
:Overload
:Overload?
:Status
:Status?
:Txfilter
:Txfilter?
:Upper
:Upper?
:TXModsens
:Lower
:Lower?
:Modlevel
:Modlevel?
:Status
:Status?
:Txfilter
:Txfilter?
:Upper
:Upper?
:TXNoise
:Averages
:Averages?
:Status
:Status?
:Txfilter
:Txfilter?
:Upper
:Upper?
:TXSInad
:Averages
:Averages?
:Lower
:Lower?
:Modlevel
:Modlevel?
:Nbmodlevel
:Nbmodlevel?
:Status
:Status?
:Txfilter
:Txfilter?
:TXSN
:Averages
:Averages?
:Lower
:Lower?
:Modlevel
:Modlevel?
:Nbmodlevel
:Nbmodlevel?
:Status
:Status?
:Txfilter
:Txfilter?

Note: The units for all response commands are in the units shown in the Example response.

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:AFFreq

:Error

Description : Sets the % error limit for the test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : PC
Example : `:PAram:AFFreq:Error 10PC`
Sets the test error limit to 10%

:AFFreq

:Error?

Description : Returns currently set value.
Response : <NR2> Range 0 to 1000
Example : `:PAram:AFFreq:Error?`
Example response : `PA:AFF:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:AFFreq

:Ref

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 500 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `:PAram:AFFreq:Ref 6kHz`
Sets the test's reference audio frequency to 6 kHz

:AFFreq

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0 to 500 kHz
Example : `:PAram:AFFreq:Ref?`
Example response : `PA:AFFr:R 6000.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:AFLevel

:Error

Description : Sets the % error limit for the test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : `PC`
Example : `:PAram:AFLevel:Error 10PC`
Sets the test error limit to 10PC

:AFLevel

:Error?

Description : Returns currently set value.
Response : <NR2> Range 0 to 1000
Example : `:PAram:AFLevel:Error?`
Example response : `PA:AFL:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:AFLevel

:Ref

Description : Sets the reference received audio level reference for the test.
Parameter : <NRf> Range 0 to 5 V RMS
Allowed suffices : `UV, MV, V`
Example : `:PAram:AFLevel:Ref 50MV`
Sets the test's reference audio level to 50 mV

:AFLevel

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0 to 5 V RMS
Example : `:PAram:AFLevel:Ref?`
Example response : `PA:AFL:R 50.0MV`

:PAram

Applicable to PMR

:Configuration

:Duplex

Description : Enables or disables duplex channels for a PMR mobile.
Parameter : <CPD> OFF, ON
Allowed suffices : Not applicable
Example : `:PAram:Configuration:Duplex ON`
Enables RF Duplex TEST mode of operation

:Configuration

:Duplex?

Description : Returns currently set value.
Response : <CRD> OFF, ON
Example : `:PAram:Configuration:Duplex?`
Example response : `PA:C:D ON`

:PAram

Applicable to GSM, DCS and PCS

:Configuration

:Mblevel

Description : Sets the mobile's power level for all tests except Power Level, Mobile Power Level and Power Profile.

Parameter : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Allowed suffices : Not applicable

Example : `:PAram:Configuration:Mblevel 9`
Sets the mobile's power level to 9 for the tests specified above

:Configuration

:Mblevel?

Description : Returns currently set value.

Response : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Example : `:PAram:Configuration:Mblevel?`

Example response : `PA:C:M 9`

:PAram

Applicable to GSM, DCS and PCS

:Configuration

:Rfgenlevel

Description : Sets the RF generator level for the following tests:-

Power Level
Tx Frequency
Tx Peak Phase Error
Tx Power Profile
Tx RMS Phase Error
Tx Timing

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:Configuration:Rfgenlevel –80 DBM`
Sets the RF generator level to –80 dBm for the tests listed above

:Configuration

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:Configuration:Rfgenlevel?`

Example response : `PA:C:R -80.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:Configuration

:TImeadvance

Description : Sets the mobile's timing advance for all tests except Timing and Timing Advance.

Parameter : <NRf> Range 0 to 63

Allowed suffices : BITS

Example : `:PAram:Configuration:TImeadvance 0BITS`
Sets the mobile's timing advance to 0 bits for the tests specified above

:Configuration

:TImeadvance?

Description : Returns currently set value.

Response : <NR1> Range 0 to 63

Example : `:PAram:Configuration:TImeadvance?`

Example response : `PA:C:TI 0.0BITS`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:Configuration

:TXsetting

Description : Sets the Tx modulation type.

Parameter : <CPD> FFT
MOD_METER

Allowed suffices : Not applicable

Example : `:PAram:Configuration:TXsetting MOD_METER`
Sets the Tx modulation type to modulation meter

:Configuration

:TXsetting?

Description : Returns currently set value.

Response : <CRD> FFT
MOD_METER

Example : `:PAram:Configuration:TXsetting?`

Example response : `PA:C:TX MOD_METER`

:PAram

Applicable to AMPS and TACS

:DATADevn

:Error

Description : Sets the % error limit for the test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : PC
Example : :PAram:DATADevn:Error 10PC
Sets the test error limit to 10%

:DATADevn

:Error?

Description : Returns currently set value.
Response : <NR2> Range 0 to 1000
Example : :PAram:DATADevn:Error?
Example response : PA:DATAD:E 10.0PC

:PAram

Applicable to AMPS and TACS

:DATADevn

:Ref

Description : Sets the reference for the FM deviation test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : :PAram:DATADevn:Ref 6kHz
Sets the test's reference FM deviation level to 6 kHz

:DATADevn

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : :PAram:DATADevn:Ref?
Example response : PA:DATAD:R 6000.0HZ

:PAram

Applicable to AMPS and TACS

:DATADevn

:Status

Description : Enables or disables the Data Deviation test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:DATADevn:Status ON`
Enables the Data Deviation test as part of an MI-BASIC test program

:DATADevn

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:DATADevn:Status?`

Example response : `PA:DATAD:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:DATAPerform

:Lower

Description : Sets the lower pass limit as a % for the test.

Parameter : <NRf> Range 0 to 100

Allowed suffices : `PC`

Example : `:PAram:DATAPerform:Lower 95PC`
Sets the test lower pass limit to 95%

:DATAPerform

:Lower?

Description : Returns currently set value.

Response : <NR2>, Range 0 to 100

Example : `:PAram:DATAPerform:Lower?`

Example response : `PA:DATAP:L 95.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:DATAPerform

:Rfgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Allowed suffices : DBM

Example : `:PAram:DATAPerform:Rfgenlevel -116dBm`
Sets the RF generator level to –116 dBm for the test

:DATAPerform

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Example : `:PAram:DATAPerform:Rfgenlevel?`

Example response : `PA:DATAP:R -116.0DBM`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:DATAPerform

:Status

Description : Enables or disables the Data Performance test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:DATAPerform:Status ON`
Enables the Data Performance test as part of an MI-BASIC test program

:DATAPerform

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:DATAPerform:Status?`

Example response : `PA:DATAP:S ON`

:PAram

Applicable to GSM, DCS and PCS

:DHandoff

:Rfgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:DHandoff:Rfgenlevel -45DBM`
Sets the RF generator level to –45 dBm for the test

:DHandoff

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:DHandoff:Rfgenlevel?`

Example response : `PA:DH:R -40.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:DHandoff

:Status

Description : Enables or disables the Digital Handoff test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:DHandoff:Status ON`
Enables the Digital Handoff test as part of an MI-BASIC test program

:DHandoff

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:DHandoff:Status?`

Example response : `PA:DH:S ON`

:PAram

Applicable to NAMPS and NTACS

:DSatdevn

:Averages

Description : Sets the number of averages for the test.
Parameter : <NR1> Range 0 to 99
Allowed suffices : Not applicable
Example : `:PAram:DSatdevn:Averages 10`
Sets the number of averages to 10

:DSatdevn

:Averages?

Description : Returns currently set value.
Response : <NR1> Range 0 to 99
Example : `:PAram:DSatdevn:Averages?`
Example response : `PA:DS:A 10`

:PAram

Applicable to NAMPS and NTACS

:DSatdevn

:Error

Description : Sets the % error limit for the test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : PC
Example : `:PAram:DSatdevn:Error 10PC`
Sets the test error limit to 10%

:DSatdevn

:Error?

Description : Returns currently set value.
Response : <NR2>, Range 0 to 1000
Example : `:PAram:DSatdevn:Error?`
Example response : `PA:DS:E 10.0PC`

:PAram

Applicable to NAMPS and NTACS.

:DSatdevn

:Ref

Description : Sets the reference narrow-band modulation level for the test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:DSatdevn:Ref 700Hz`
Sets the test's reference modulation level to 700 Hz

:DSatdevn

:Ref?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:DSatdevn:Ref?`

Example response : `PA:DS:R 700.0HZ`

:PAram

Applicable to NAMPS and NTACS.

:DSatdevn

:Status

Description : Enables or disables the Digital SAT test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:DSatdevn:Status ON`
Enables the Digital SAT test as part of an MI-BASIC test program

:DSatdevn

:Status?

Description : Returns currently set value.

Response : <CRD> OFF or ON

Example : `:PAram:DSatdevn:Status?`

Example response : `PA:DS:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and PMR

:DTmfdecode

:Accport

Description : Assigns the active accessory port line for the test.

Parameter : <CPD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3

Allowed suffices : Not applicable

Example : `:PAram:DTmfdecode:Accport LOGIC_1`
Sets accessory port line 1 active for the DTMF test

:DTmfdecode

:Accport?

Description : Returns currently set value.

Response : <CRD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3

Example : `:PAram:DTmfdecode:Accport?`

Example response : `PA:DT:A LOGIC_1`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and PMR

:DTmfdecode

:Status

Description : Enables or disables the DTMF test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:DTmfdecode:Status ON`
Enables the DTMF test as part of an MI-BASIC test program.

:DTmfdecode

:Status?

Description : Returns currently set value.

Response : <CRD> OFF or ON

Example : `:PAram:DTmfdecode:Status?`

Example response : `PA:DT:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and PMR

:DTmfdecode

:Timeout

Description : Sets the response timeout limit for the test.

Parameter : <NR2> Range 0.001 to 600

Allowed suffices : S

Example : `:PAram:DTmfdecode:Timeout 20S`
Sets the test timeout to 20 seconds before a response

:DTmfdecode

:Timeout?

Description : Returns currently set value.

Response : <NR2> Range 0.001 to 600

Example : `:PAram:DTmfdecode:Timeout?`

Example response : `PA:DT:T 20.0S`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and PMR

:DTmfdecode

:Useaccport

Description : Enables or disables use of the accessory port logic lines as part of the DTMF test.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:DTmfdecode:Useaccport ON`
Enables use of the accessory port as part of the DTMF test

:DTmfdecode

:Useaccport?

Description : Returns currently set value.

Response : <CRD> OFF or ON

Example : `:PAram:DTmfdecode:Useaccport?`

Example response : `PA:DT:U ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:Fmdevn

:Error

Description : Sets the % error limit for the test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : PC
Example : :PAram:Fmdevn:Error 10PC
Sets the test error limit to 10%

:Fmdevn

:Error?

Description : Returns currently set value.
Response : <NR2> Range 0 to 1000
Example : :PAram:Fmdevn:Error?
Example response : PA:F:E 10.0PC

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:Fmdevn

:Ref

Description : Sets the reference for the FM deviation test.
Parameter : <NRf>, Range 0 to 40 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : :PAram:Fmdevn:Ref 6kHz
Sets the test's reference FM deviation level to 6 kHz

:Fmdevn

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : :PAram:Fmdevn:Ref?
Example response : PA:F:R 6000.0HZ

:PAram

Applicable to NAMPS and NTACS

:HAndoff

:Chantype

Description : Specifies the type of channel to which the test is to handoff.

Parameter : <CPD> NARROW
WIDE
ROTATE
LAST
ABOVE (NAMPS only)
BELOW (NAMPS only)

Allowed suffices : Not applicable

Example : `:PAram:HAndoff:Chantype NARROW`
Sets the handoff channel type to be narrow-band

:HAndoff

:Chantype?

Description : Returns currently set value.

Response : <CRD> NARROW
WIDE
ROTATE
LAST
ABOVE (NAMPS only)
BELOW (NAMPS only)

Example : `:PAram:HAndoff:Chantype?`

Example response : `PA:HA:C NARROW`

:PAram

Applicable to NMT

:HAndoff

:Mblevel

Description : Sets the power level for mobile transmissions for the handoff test.

Parameter : <NR1> Range 0 to 3

Allowed suffices : Not applicable

Example : `:PAram:HAndoff:Mblevel 1`
The NMT mobile will now be forced to transmit at power level 1 when the test is executed

:HAndoff

:Mblevel?

Description : Returns currently set value.

Response : <NR1> Range 0 to 3.

Example : `:PAram:HAndoff:Mblevel?`

Example response : `PA:HA:M 1`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT

:HAndoff

:Rfgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Allowed suffices : `DBM`

Example : `:PAram:HAndoff:Rfgenlevel -40DBM`
Sets the RF generator level to –40 dBm for the test

:HAndoff

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range -135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Example : `:PAram:HAndoff:Rfgenlevel?`

Example response : `PA:HA:R -40.0DBM`

:PAram

Applicable to MPT1327

:HAndoff

:Timeout

Description : Sets the response timeout limit for the test.
Parameter : <NR2> Range 0.001 to 600
Allowed suffices : S
Example : `:PAram:HAndoff:Timeout 20S`
Sets the test timeout to 20 seconds before a response

:HAndoff

:Timeout?

Description : Returns currently set value.
Response : <NR2> Range 0.001 to 600
Example : `:PAram:HAndoff:Timeout?`
Example response : `PA:HA:T 20.0S`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:HOokflash

:Accport

Description : Assigns the active accessory port line for the test.
Parameter : <CPD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3
Allowed suffices : Not applicable
Example : `:PAram:HOokflash:Accport LOGIC_1`
Sets accessory port line 1 active for the hookflash test

:HOokflash

:Accport?

Description : Returns currently set value.
Response : <CRD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3
Example : `:PAram:HOokflash:Accport?`
Example response : `PA:HO:A LOGIC_1`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:HOokflash

:Status

Description : Enables or disables the hookflash test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:HOokflash:Status ON`
Enables the hookflash test as part of an MI-BASIC test program

:HOokflash

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:HOokflash:Status?`

Example response : `PA:HO:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:HOokflash

:Timeout

Description : Sets the response timeout limit for the test.

Parameter : <NR2> Range 0.001 to 600

Allowed suffices : `S`

Example : `:PAram:HOokflash:Timeout 20S`
Sets the test timeout to 20 seconds before a response

:HOokflash

:Timeout?

Description : Returns currently set value.

Response : <NR2> Range 0.001 to 600

Example : `:PAram:HOokflash:Timeout?`

Example response : `PA:HO:T 20.0S`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:HOokflash

:Useaccport

Description : Enables or disables use of the accessory port logic lines as part of the DTMF test.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:HOokflash:Useaccport ON`
Enables use of the accessory port as part of the hookflash test

:HOokflash

:Useaccport?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:HOokflash:Useaccport?`

Example response : `PA:HO:U ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:Landclear

:Rfgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Allowed suffices : DBM

Example : `:PAram:Landclear:Rfgenlevel -40DBM`
Sets the RF generator level to –40 dBm for the test

:Landclear

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Example : `:PAram:Landclear:Rfgenlevel?`

Example response : `PA:L:R -40.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:Landclear

:Status

Description : Enables or disables the Clear From Land test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:Landclear:Status ON`
Enables the Clear From Land test as part of an MI-BASIC test program

:Landclear

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:Landclear:Status?`

Example response : `PA:L:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:MOBileclear

:Accport

Description : Assigns the active accessory port line for the test.

Parameter : <CPD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3

Allowed suffices : Not applicable

Example : `:PAram:MOBileclear:Accport LOGIC_1`
Sets accessory port line 1 active for the Clear From Mobile test

:MOBileclear

:Accport?

Description : Returns currently set value.

Response : <CRD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3

Example : `:PAram:MOBileclear:Accport?`

Example response : `PA:MOB:A LOGIC_1`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:MOBileclear

:Rfgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Allowed suffices : DBM

Example : `:PAram:MOBileclear:Rfgenlevel -40DBM`
Sets the RF generator level to –40 dBm for the test

:MOBileclear

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Example : `:PAram:MOBileclear:Rfgenlevel?`

Example response : `PA:MOB:R -40.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:MOBileclear

:Status

Description : Enables or disables the Clear From Mobile test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:MOBileclear:Status ON`
Enables the Clear From Mobile test as part of an MI-BASIC test program

:MOBileclear

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:Landclear:Status?`

Example response : `PA:MOB:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:MOBileclear

:Timeout

Description : Sets the response timeout limit for the test.
Parameter : <NR2> Range 0.001 to 600
Allowed suffices : S
Example : `:PAram:MOBileclear:Timeout 20S`
Sets the test timeout to 20 seconds before a response

:MOBileclear

:Timeout?

Description : Returns currently set value.
Response : <NR2> Range 0.001 to 600
Example : `:PAram:MOBileclear:Timeout?`
Example response : `PA:MOB:T 20.0S`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:MOBileclear

:Useaccport

Description : Enables or disables use of the accessory port logic lines as part of the Clear From Mobile test.
Parameter : <CPD> OFF, ON
Allowed suffices : Not applicable
Example : `:PAram:MOBileclear:Useaccport ON`
Enables use of the accessory port as part of the Clear From Mobile test

:MOBileclear

:Useaccport?

Description : Returns currently set value.
Response : <CRD> OFF, ON
Example : `:PAram:MOBileclear:Useaccport?`
Example response : `PA:MOB:U ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:MODfreq

:Error

Description : Sets the % error limit for the test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : PC
Example : `:PAram:MODfreq:Error 10PC`
Sets the test error limit to 10%

:MODfreq

:Error?

Description : Returns currently set value.
Response : <NR2> Range 0 to 1000
Example : `:PAram:MODfreq:Error?`
Example response : `PA:MOD:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:MODfreq

:Ref

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 20 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : `PAram:MODfreq:Ref 2.3kHz`
Sets the test's reference modulation frequency to 2.3 kHz

:MODfreq

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0 to 20 kHz
Example : `:PAram:MODfreq:Ref?`
Example response : `PA:MOD:R 2300.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PAgemobile

:Accport

Description : Assigns the active accessory port line for the test.

Parameter : <CPD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3

Allowed suffices : Not applicable

Example : `:PAram:PAgemobile:Accport LOGIC_1`
Sets accessory port line 1 active for the Page Mobile test

:PAgemobile

:Accport?

Description : Returns currently set value.

Response : <CRD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3

Example : `:PAram:PAgemobile:Accport?`

Example response : `PA:PA:A LOGIC_1`

:PAram

Applicable to NAMPS and NTACS

:PAgemobile

:Chantype

Description : Specifies the type of voice or traffic channel the Page Mobile test is to establish.

Parameter : <CPD> NARROW
WIDE
ROTATE
ABOVE (NAMPS only)
BELOW (NAMPS only)

Allowed suffices : Not applicable

Example : `:PAram:PAgemobile:Chantype NARROW`
Sets the voice channel type to be narrow-band when the call is answered

:PAgemobile

:Chantype?

Description : Returns currently set value.

Response : <CRD> NARROW
WIDE
ROTATE
ABOVE (NAMPS only)
BELOW (NAMPS only)

Example : `:PAram:PAgemobile:Chantype?`

Example response : `PA:PA:C NARROW`

:PAram

Applicable to NMT

:PAgemobile

:Mblevel

Description : Sets the power level for mobile transmissions for the Page Mobile test.

Parameter : <NR1> Range 0 to 3

Allowed suffices : Not applicable

Example : `:PAram:PAgemobile:Mblevel 1`
The NMT mobile will now be forced to transmit at power level 1 when the test is executed

:PAgemobile

:Mblevel?

Description : Returns currently set value.

Response : <NR1> Range 0 to 3.

Example : `:PAram:PAgemobile:Mblevel?`

Example response : `PA:PA:M 1`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PAgemobile

:Rfgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Allowed suffices : `DBM`

Example : `:PAram:PAgemobile:Rfgenlevel -40DBM`
Sets the RF generator level to –40 dBm for the test

:PAgemobile

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Example : `:PAram:PAgemobile:Rfgenlevel?`

Example response : `PA:PA:R -40.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:PAgemobile

:Status

Description : Enables or disables the Page Mobile test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:PAgemobile:Status ON`
Enables the Page Mobile test as part of an MI-BASIC test program

:PAgemobile

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:PAgemobile:Status?`

Example response : `PA:PA:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PAgemobile

:Timeout

Description : Sets the response timeout limit for the test.

Parameter : <NR2> Range 0.001 to 600

Allowed suffices : `S`

Example : `:PAram:PAgemobile:Timeout 20S`
Sets the test timeout to 20 seconds before a response

:PAgemobile

:Timeout?

Description : Returns currently set value.

Response : <NR2> Range 0.001 to 600

Example : `:PAram:PAgemobile:Timeout?`

Example response : `PA:PA:T 20.0S`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PAgemobile

:Useaccport

Description : Enables or disables use of the accessory port logic lines as part of the Page Mobile test.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:PAgemobile:Useaccport ON`
Enables use of the accessory port as part of the Page Mobile test

:PAgemobile

:Useaccport?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:PAgemobile:Useaccport?`

Example response : `PA:PA:U ON`

:PAram

Applicable to GSM, DCS and PCS

:PKPhase

:Bursttst

Description : Sets the number of bursts tested by the test.

Parameter : <NR1> Range 1 to 100

Allowed suffices : Not applicable

Example : `:PAram:PKPhase:Bursttst 20`
Sets the number of bursts tested to 20 for the test

:PKPhase

:Bursttst?

Description : Returns current set value for the test.

Response : <NR1> Range 1 to 100

Example : `:PAram:PKPhase:Bursttst?`

Example response : `PA:PKP:B 20`

:PAram

Applicable to GSM, DCS and PCS

:PKPhase

:Meastype

Description : Defines the type of measurement for the test.

Parameter : <CPD> WORSTCASE
AVERAGE

Allowed suffices : Not applicable

Example : `:PAram:PKPhase:Meastype WORSTCASE`
Sets the test's measurement type to worst case

:PKPhase

:Meastype?

Description : Returns currently set value for the test

Parameter : <CRD> WORSTCASE
AVERAGE

Example : `:PAram:PKPhase:Meastype?`

Example Response : `:PA:PKP:M WORSTCASE`

:PAram

Applicable to GSM, DCS and PCS

:PKPhase

:Status

Description : Enables or disables the Tx Peak Phase Error test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:PKPhase:Status ON`
Enables the Tx Peak Phase Error test as part of an MI-BASIC test program

:PKPhase

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:PKPhase:Status?`

Example response : `PA:PKP:S ON`

:PAram

Applicable to GSM, DCS and PCS

:PKPhase

:Upper

Description : Sets the upper phase error for the test.
Parameter : <NRf> Range 0 to 30
Allowed suffices : DEG
Example : `:PAram:PKPhase:Upper 20DEG`
Sets the upper phase error to 20° for the test

:PKPhase

:Upper?

Description : Returns currently set value.
Response : <NR2> Range 0 to 30
Example : `:PAram:PKPhase:Upper?`
Example response : `PA:PKP:U 20.0DEG`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PLacecall

:Accport

Description : Assigns the active accessory port line for the test.
Parameter : <CPD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3
Allowed suffices : Not applicable
Example : `:PAram:PLacecall:Accport LOGIC_1`
Sets accessory port line 1 active for the Place Call test

:PLacecall

:Accport?

Description : Returns currently set value.
Response : <CRD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3
Example : `:PAram:PLacecall:Accport?`
Example response : `PA:PL:A LOGIC_1`

:PAram

Applicable to NAMPS and NTACS

:PLacecall

:Chantype

Description : Specifies the type of voice or traffic channel the Place Call test is to establish.

Parameter : <CPD> NARROW
WIDE
ROTATE
ABOVE (NAMPS only)
BELOW (NAMPS only)

Allowed suffices : Not applicable

Example : `:PAram:PLacecall:Chantype NARROW`
Sets the voice channel type to be narrow-band when the call is answered

:PLacecall

:Chantype?

Description : Returns currently set value.

Response : <CRD> NARROW
WIDE
ROTATE
ABOVE (NAMPS only)
BELOW (NAMPS only)

Example : `:PAram:PLacecall:Chantype?`

Example response : `PA:PL:C NARROW`

:PAram

Applicable to NMT

:PLacecall

:Mblevel

Description : Sets the power level for mobile transmissions for the Place Call test.

Parameter : <NR1> Range 0 to 3

Allowed suffices : Not applicable

Example : `:PAram:PLacecall:Mblevel 1`
The NMT mobile will now be forced to transmit at power level 1 when the test is executed

:PLacecall

:Mblevel?

Description : Returns currently set value.

Response : <NR1> Range 0 to 3.

Example : `:PAram:PLacecall:Mblevel?`

Example response : `PA:PL:M 1`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PLacecall

:Rfgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Allowed suffices : `DBM`

Example : `:PAram:PLacecall:Rfgenlevel -40DBM`
Sets the RF generator level to –40 dBm for the test

:PLacecall

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Example : `:PAram:PLacecall:Rfgenlevel?`

Example response : `PA:PL:R -40.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:PLacecall

:Status

Description : Enables or disables the Place Call test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example :
```
:PAram:PLacecall:Status ON
```
Enables the Place Call test as part of an MI-BASIC test program

:PLacecall

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example :
```
:PAram:PLacecall:Status?
```

Example response :
```
PA:PL:S ON
```

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PLacecall

:Timeout

Description : Sets the response timeout limit for the test.

Parameter : <NR2> Range 0.001 to 600

Allowed suffices : S

Example :
```
:PAram:PLacecall:Timeout 20S
```
Sets the test timeout to 20 seconds before a response

:PLacecall

:Timeout?

Description : Returns currently set value.

Response : <NR2> Range 0.001 to 600

Example :
```
:PAram:PLacecall:Timeout?
```

Example response :
```
PA:PL:T 20.0S
```

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:PLacecall

:Useaccport

Description : Enables or disables use of the accessory port logic lines as part of the Place Call test.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:PLacecall:Useaccport ON`
Enables use of the accessory port as part of the Place Call test

:PLacecall

:Useaccport?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:PLacecall:Useaccport?`

Example response : `PA:PL:U ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and NMT

:POwerlevel

:Averages

Description : Sets the number of averages for the test.

Parameter : <NR1> Range 0 to 99

Allowed suffices : Not applicable

Example : `:PAram:POwerlevel:Averages 10`
Sets the number of averages to 10

:POwerlevel

:Averages?

Description : Returns currently set value.

Response : <NR1> Range 0 to 99

Example : `:PAram:POwerlevel:Averages?`

Example response : `PA:PO:A 10`

:PAram

Applicable to GSM, DCS and PCS

:POwerlevel

:Firstvalue

Description : Sets the first power level to test as part of an MI-BASIC test program.

Parameter : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Allowed suffices : Not applicable

Example : `:PAram:POwerlevel:Firstvalue 2`
Sets the first power level to test in an MI-BASIC test program to 2

:POwerlevel

:Firstvalue?

Description : Returns currently set value.

Response : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Example : `:PAram:POwerlevel:Firstvalue?`

Example response : `PA:PO:F 2`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, GSM, DCS and PCS

:POwerlevel

:HIRLower

Description : Sets the lower limit for the highest power level transmitted by the mobile and for power level 30 for PCS.

Parameter : <NRf> Range –100 to 100

Allowed suffices : `DB`

Example : `:PAram:POwerlevel:HIRLower 2dB`
Sets the higher power level lower limit to 2 dB

:POwerlevel

:HIRLower?

Description : Returns currently set value.

Response : <NR2> Range –100 to 100

Example : `:PAram:POwerlevel:HIRLower?`

Example response : `PA:PO:HIRL 2.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, GSM, DCS and PCS

:POwerlevel

:HIRUpper

Description : Sets the upper limit for the highest power level transmitted by the mobile and for power level 30 for PCS.

Parameter : <NRf> Range –100 to 100

Allowed suffices : DB

Example : `:PAram:POwerlevel:HIRUpper 2dB`
Sets the highest power level upper limit to 2 dB

:POwerlevel

:HIRUpper?

Description : Returns currently set value.

Response : <NR2> Range –100 to 100

Example : `:PAram:POwerlevel:HIRUpper?`

Example response : `PA:PO:HIRU 2.0DB`

:PAram

Applicable to GSM, DCS and PCS

:POwerlevel

:Increment

Description : Sets the value by which to increment the power level for the test as part of an MI-BASIC test program.

Parameter : <NR1> Range 0-19 for GSM, 0-15 for DCS and PCS

Allowed suffices : Not applicable

Example : `:PAram:POwerlevel:Increment 2`
Sets the increment value to 2

:POwerlevel

:Increment?

Description : Returns currently set value.

Response : <NR1> Range 0-19 for GSM, 0-15 for DCS and PCS

Example : `:PAram:POwerlevel:Increment?`

Example response : `PA:PO:I 2`

:PAram

Applicable to GSM, DCS and PCS

:POwerlevel

:LAstvalue

Description : Sets the last power level to test as part of an MI-BASIC test program.

Parameter : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Allowed suffices : Not applicable

Example : `:PAram:POwerlevel:LAstvalue 15`
Sets the last power level to test in an MI-BASIC test program to 15

:POwerlevel

:LAstvalue?

Description : Returns currently set value.

Response : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Example : `:PAram:POwerlevel:LAstvalue?`

Example response : `PA:PO:LA 15`

:PAram

Applicable to GSM, DCS and PCS

:POwerlevel

:LORLower

Description : Sets the lower limit for power levels 16-19 for GSM, or 14-15 for DCS/PCS, transmitted by the mobile during the test.

Parameter : <NRf> Range –100 to 100

Allowed suffices : `DB`

Example : `:PAram:POwerlevel:LORLower 5DB`
Sets the low power level lower limit to 5 dB

:POwerlevel

:LORLower?

Description : Returns currently set value.

Response : <NR2> Range –100 to 100

Example : `:PAram:POwerlevel:LORLower?`

Example response : `PA:PO:LORL 5.0DB`

:PAram

Applicable to GSM, DCS and PCS

:POwerlevel

:LORUpper

Description : Sets the upper limit for power levels 16-19 for GSM, or 14-15 for DCS/PCS, transmitted by the mobile during the test.

Parameter : <NRf> Range –100 to 100

Allowed suffices : DB

Example : `:PAram:POwerlevel:LORUpper 5DB`
Sets the low power level upper limit to 5 dB

:POwerlevel

:LORUpper?

Description : Returns currently set value.

Response : <NR2> Range –100 to 100

Example : `:PAram:POwerlevel:LORUpper?`

Example response : `PA:PO:LORU 5.0DB`

:PAram

Applicable to DCS and PCS

:POwerlevel

:MIDRLower

Description : Sets the lower limit for power levels 9-13.

Parameter : <NRf> Range –100 to 100

Allowed suffices : DB

Example : `:PAram:POwerlevel:MIDRLower 4.0DB`
Sets the mid power level lower limit to 4 dB

:POwerlevel

:MIDRLower?

Description : Returns currently set value.

Parameter : <NR2> Range –100 to 100

Allowed suffices : DB

Example : `:PAram:POwerlevel:MIDRLower?`

Example response `PA:PO:MIDRL 4.0DB`

:PAram

Applicable to DCS and PCS

:POwerlevel

:MIDRUpper?

Description : Sets the upper limit for power levels 9-13.

Parameter : <NRf> Range –100 to 100

Allowed suffices : DB

Example : `:PAram:POwerlevel:MIDRUpper 4.0DB`
Sets the mid power level upper limit to 4 dB

:POwerlevel

:MIDRUpper?

Description : Returns currently set value.

Response : <NR2> Range –100 to 100

Allowed suffices : DB

Example : `:PAram:POwerlevel:MIDRUpper?`

Example response : `PA:PO:MIDRU 4.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, GSM, DCS and PCS

:POwerlevel

:RLower

Description : Sets the lower limit for the power levels transmitted by the mobile during the test other than those set by :HIRLower, :MIDLower and :LORLower.

Parameter : <NRf> Range –100 to 100

Allowed suffices : DB

Example : `:PAram:POwerlevel:RLower 2dB`
Sets the power level lower limit to 2 dB

:POwerlevel

:RLower?

Description : Returns currently set value.

Response : <NR2> Range –100 to 100

Example : `:PAram:POwerlevel:RLower?`

Example response : `PA:PO:RL 2.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, GSM, DCS and PCS

:POwerlevel

:RUpper

Description : Sets the upper limit for the power levels transmitted by the mobile during the test other than those set by :HIRUpper, :MIDUpper and :LORUpper.

Parameter : <NRf> Range –100 to 100

Allowed suffices : DB

Example : `:PAram:POwerlevel:RUpper 2dB`
Sets the power level lower limit to 2 dB

:POwerlevel

:RUpper?

Description : Returns currently set value.

Response : <NR2> Range –100 to 100

Example : `:PAram:POwerlevel:RUpper?`

Example response : `PA:PO:RU 2.0DBM`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, GSM, DCS and PCS

:POwerlevel

:Status

Description : Enables or disables the Power Level test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:POwerlevel:Status ON`
Enables the Power Level test as part of an MI-BASIC test program

:POwerlevel

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:POwerlevel:Status?`

Example response : `PA:PO:S ON`

:PAram

Applicable to GSM, DCS and PCS

:PRofile

:Firstvalue

Description : Sets the first power level to test as part of an MI-BASIC test program.

Parameter : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Allowed suffices : Not applicable

Example : `:PAram:PRofile:Firstvalue 2`
Sets the first power level to be tested to 2 for the test as part of an MI-BASIC test program

:PRofile

:Firstvalue?

Description : Returns currently set value.

Response : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Example : `:PAram:PRofile:Firstvalue?`

Example response : `PA:PR:F 2`

:PAram

Applicable to GSM, DCS and PCS

:PRofile

:Increment

Description : Sets the value by which to increment the power level for the test as part of an MI-BASIC test program.

Parameter : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Allowed suffices : Not applicable

Example : `:PAram:PRofile:Increment 7`
Sets the increment value to 7

:PRofile

:Increment?

Description : Returns currently set value.

Response : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Example : `:PAram:PRofile:Increment?`

Example response : `PA:PR:I 7`

:PAram

Applicable to GSM, DCS and PCS

:PRofile

:Lastvalue

Description : Sets the last power level to test as part of an MI-BASIC test program.

Parameter : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Allowed suffices : Not applicable

Example : `:PAram:PRofile:Lastvalue 15`
Sets the last power level to be tested to 15 for the test as part of an MI-BASIC test program

:PRofile

:Lastvalue?

Description : Returns currently set value.

Response : <NR1> Range 0-19 for GSM, 29-31 and 0-15 for DCS, or 30-31 and 0-15 for PCS

Example : `:PAram:PRofile:Lastvalue?`

Example response : `PA:PR:L 15`

:PAram

Applicable to GSM, DCS and PCS

:PRofile

:Status

Description : Enables or disables the Power Profile test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:PRofile:Status ON`
Enables the Power Profile test as part of an MI-BASIC test program

:PRofile

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:PRofile:Status?`

Example response : `PA:PR:S ON`

:PAram

Applicable to MPT1327 and PMR

:PTTOFf

:Lref

Description : Sets the measured RF level reference below which the mobile is deemed to have had its PTT switch released.

Parameter : <NRf> Range –135 to 48.75

Allowed suffices : DBM

Example : `:PAram:PTTOFf:Lref -20dBm`
Sets the 'PTT is off' reference to –20 dBm

:PTTOFf

:Lref?

Description : Returns currently set value.

Response : <NR2> Range –135 to 48.75

Example : `:PAram:PTTOFf:Lref?`

Example response : `PA:PTTOF:L -20.0DBM`

:PAram

Applicable to MPT1327 and PMR

:PTTOFf

:Timeout

Description : Sets the response timeout limit for the test.

Parameter : <NR2> Range 0.001 to 600

Allowed suffices : S

Example : `:PAram:PTTOFf:Timeout 20S`
Sets the test timeout to 20 seconds before a response

:PTTOFf

:Timeout?

Description : Returns currently set value.

Response : <NR2> Range 0.001 to 600

Example : `:PAram:PTTOFf:Timeout?`

Example response : `PA:PTTOF:T 20.0S`

:PAram

Applicable to MPT1327

:PTTOFf

:Usepressel

Description : Enables or disables Pressel confirmation messages.
Parameter : <CPD> ON, OFF
Allowed suffices : Not applicable
Example : `:PAram:PTTOFf:Usepressel ON`
Sets the test to expect 'PRESSEL OFF' messages from the mobile

:PTTOFf

:Usepressel?

Description : Returns currently set value.
Response : <CRD> ON, OFF
Example : `:PAram:PTTOFf:Usepressel?`
Example response : `PA:PTTOF:U ON`

:PAram

Applicable to MPT1327 and PMR

:PTTON

:Accport

Description : Assigns the active accessory port line for the test.
Parameter : <CPD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3
Allowed suffices : Not applicable
Example : `:PAram:PTTON:Accport LOGIC_1`
Sets accessory port line 1 active for the PTTON test

:PTTON

:Accport?

Description : Returns currently set value.
Response : <CRD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3
Example : `:PAram:PTTON:Accport?`
Example response : `PA:PTTON:A LOGIC_1`

:PAram

Applicable to MPT1327 and PMR

:PTTON

:Lref

Description : Sets the measured RF level reference above which the mobile is deemed to have had its PTT switch pressed.

Parameter : <NRf> Range –135 to 48.75

Allowed suffices : DBM

Example : `:PAram:PTTON:Lref -20dBm`
Sets the 'PTT is on' reference to –20 dBm

:PTTON

:Lref?

Description : Returns currently set value.

Response : <NR2> Range –135 to 48.75

Example : `:PAram:PTTON:Lref?`

Example response : `PA:PTTON:L -20.0DBM`

:PAram

Applicable to MPT1327 and PMR

:PTTON

:Timeout

Description : Sets the response timeout limit for the test.

Parameter : <NR2> Range 0.001 to 600

Allowed suffices : S

Example : `:PAram:PTTON:Timeout 20S`
Sets the test timeout to 20 seconds before a response

:PTTON

:Timeout?

Description : Returns currently set value.

Response : <NR2> Range 0.001 to 600

Example : `:PAram:PTTON:Timeout?`

Example response : `PA:PTTON:T 20.0S`

:PAram

Applicable to MPT1327 and PMR

:PTTON

:USEAccport

Description : Enables or disables use of the accessory port logic lines as part of the PTTON test.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:PTTON:USEAccport ON`
Enables use of the accessory port as part of the PTTON test

:PTTON

:USEAccport?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:PTTON:USEAccport?`

Example response : `PA:PTTON:USEA ON`

:PAram

Applicable to MPT1327

:PTTON

:USEPressel

Description : Enables or disables Pressel confirmation messages.

Parameter : <CPD> ON, OFF

Allowed suffices : Not applicable

Example : `:PAram:PTTON:USEPressel ON`
Sets the test to expect 'PRESSEL ON' messages from the mobile

:PTTON

:USEPressel?

Description : Returns currently set value.

Response : <CRD> ON, OFF

Example : `:PAram:PTTON:USEPressel?`

Example response : `PA:PTTON:USEP ON`

:PAram

Applicable to NMT

:REGIster

:ACcport

Description : Assigns the active accessory port line for the test.
Parameter : <CPD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3
Allowed suffices : Not applicable
Example : `:PAram:REGIster:ACcport LOGIC_1`
Sets accessory port line 1 active for the Register test

:REGIster

:ACcport?

Description : Returns currently set value.
Response : <CRD> LOGIC_0, LOGIC_1, LOGIC_2, and LOGIC_3
Example : `:PAram:REGIster:ACcport?`
Example response : `PA:REGI:AC LOGIC_1`

:PAram

Applicable to GSM, DCS and PCS

:REGIster

:AUthchk

Description : Enables or disables Authentication Check as part of the test.
Parameter : <CPD> OFF, ON
Allowed suffices : Not applicable
Example : `:PAram:REGIster:AUthchk ON`
Enables Authentication Check as part of the test

:REGIster

:AUthchk?

Description : Returns currently set value.
Response : <CRD> OFF, ON
Example : `:PAram:REGIster:AUthchk?`
Example response : `PA:REGI:AU ON`

:PAram

Applicable to NMT

:REGIster

:Mblevel

Description : Sets the power level for mobile transmissions for the Register test.
Parameter : <NR1> Range 0 to 3
Allowed suffices : Not applicable
Example : `:PAram:REGIster:Mblevel 1`
The NMT mobile will be forced to transmit at power level 1 when the test is executed

:REGIster

:Mblevel?

Description : Returns currently set value.
Response : <NR1> Range 0 to 3.
Example : `:PAram:REGIster:Mblevel?`
Example response : `PA:REGI:M 1`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, GSM, DCS and PCS

:REGIster

:REtryenabled

Description : Enables or disables retrying as part of the Register test.
Parameter : <CPD> ON, OFF
Allowed suffices : Not applicable
Example : `:PAram:REGIster:REtryenabled ON`
Enables retry as part of the Register test

:REGIster

:REtryenabled?

Description : Returns currently set value.
Response : <CRD> ON, OFF
Example : `:PAram:REGIster:REtryenabled?`
Example response : `PA:REGI:RE ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:REGIster

:RFgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Allowed suffices : DBM

Example : `:PAram:REGIster:RFgenlevel –40DBM`
Sets the RF generator level to –40 dBm for the test

:REGIster

:RFgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Example : `:PAram:REGIster:RFgenlevel?`

Example response : `PA:REGI:RF –40.0DBM`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS and PCS

:REGIster

:Status

Description : Enables or disables the Register test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:REGIster:Status ON`
Enables the Register test as part of an MI-BASIC test program

:REGIster

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:REGIster:Status?`

Example response : `PA:REGI:S ON`

:PAram

Applicable to NMT, GSM, DCS and PCS

:REGIster

:Timeout

Description : Sets the response timeout limit for the test.

Parameter : <NR2> Range 0.001 to 600

Allowed suffices : S

Example : `:PAram:REGIster:Timeout 35S`
Sets the test timeout to 35 seconds before a response

:REGIster

:Timeout?

Description : Returns currently set value.

Response : <NR2> Range 0.001 to 600

Example : `:PAram:REGIster:Timeout?`

Example response : `PA:REGI:T 35.0S`

:PAram

Applicable to NMT

:REGIster

:Useaccport

Description : Enables or disables use of the accessory port logic lines as part of the Register test.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:REGIster:Useaccport ON`
Enables use of the accessory port as part of the Register test

:REGIster

:Useaccport?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:REGIster:Useaccport?`

Example response : `PA:REGI:U ON`

:PAram

Applicable to GSM, DCS and PCS

:REGWarp

:AUthchk

Description : Enables or disables Authentication Check as part of the test.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:REGWarp:AUthchk ON`
Enables Authentication Check as part of the test

:REGWarp

:AUthchk?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:REGWarp:AUthchk?`

Example response : `PA:REGW:AU ON`

:PAram

Applicable to GSM, DCS and PCS

:REGWarp

:REtryenabled

Description : Enables or disables retrying as part of the Registration Warp test.

Parameter : <CPD> ON, OFF

Allowed suffices : Not applicable

Example : `:PAram:REGWarp:REtryenabled ON`
Enables retry as part of the Registration Warp test

:REGWarp

:REtryenabled?

Description : Returns currently set value.

Response : <CRD> ON, OFF

Example : `:PAram:REGWarp:REtryenabled?`

Example response : `PA:REGW:RE ON`

:PAram

Applicable to GSM, DCS and PCS

:REGWarp

:RFgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:REGWarp:RFgenlevel –40DBM`
Sets the RF generator level to –40 dBm for the test

:REGWarp

:RFgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:REGWarp:RFgenlevel?`

Example response : `PA:REGW:RF –40.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:REGWarp

:Status

Description : Enables or disables the Registration Warp test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:REGWarp:Status ON`
Enables the Registration Warp test as part of an MI-BASIC test program

:REGWarp

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:REGWarp:Status?`

Example response : `PA:REGW:S ON`

:PAram

Applicable to GSM, DCS and PCS

:REGWarp

:Timeout

Description : Sets the response timeout limit for the test.

Parameter : <NR2> Range 0.001 to 600

Allowed suffices : S

Example : `:PAram:REGWarp:Timeout 45S`
Sets the test timeout to 45 seconds before a response

:REGWarp

:Timeout?

Description : Returns currently set value.

Response : <NR2> Range 0.001 to 600

Example : `:PAram:REGWarp:Timeout?`

Example response : `PA:REGW:T 45.0S`

:PAram

Applicable to GSM, DCS and PCS

:REGWarp

:Warptype

Description : Defines the type of register warp test performed as part of an MI-BASIC test program

Parameter : <CPD> LOWHIGH
HIGH
LOW

Allowed suffices : Not applicable

Example : `:PAram:REGWarp:Warptype LOWHIGH`
Allows a high and low register warp test to be performed as part of an MI-BASIC test program.

:REGWarp

:Warptype?

Description : Returns currently set value for the test

Parameter : <CRD> LOWHIGH
HIGH
LOW

Example : `:PAram:REGWarp:Warptype?`

Example response : `:PA:REGW:W LOWHIGH`

:PAram

Applicable to GSM, DCS and PCS

:RMSPhase

:Bursttst

Description : Sets the number of bursts tested by the test.

Parameter : <NR1> Range 1 to 100

Allowed suffices : Not applicable

Example : `:PAram:RMSPhase:Bursttst 20`
Sets the number of bursts tested by the test to 20

:RMSPhase

:Bursttst?

Description : Returns currently set value.

Response : <NR1> Range 1 to 100

Example : `:PAram:RMSPhase:Bursttst?`

Example response : `PA:RMSP:B 20`

:PAram

Applicable to GSM, DCS and PCS

:RMSPhase

:Meastype

Description : Defines the type of measurement for the test.

Parameter : <CPD> WORSTCASE
AVERAGE

Allowed suffices : Not applicable

Example : `:PAram:RMSPhase:Meastype WORSTCASE`
Sets the test's measurement type to worst case

:RMSPhase

:Meastype?

Description : Returns currently set value for the test

Parameter : <CRD> WORSTCASE
AVERAGE

Example : `:PAram:RMSPhase:Meastype?`

Example response : `:PA:RMSP:M WORSTCASE`

:PAram

Applicable to GSM, DCS and PCS

:RMSPhase

:Status

Description : Enables or disables the Tx RMS Phase Error test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RMSPhase:Status ON`
Enables the Tx RMS Phase Error test as part of an MI-BASIC test program

:RMSPhase

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RMSPhase:Status?`

Example response : `PA:RMSP:S ON`

:PAram

Applicable to GSM, DCS and PCS

:RMSPhase

:Upper

Description : Sets the upper phase error for the test.

Parameter : <NRf> Range 0 to 30

Allowed suffices : DEG

Example : `:PAram:RMSPhase:Upper 5DEG`
Sets the upper phase error for the test to 5°

:RMSPhase

:Upper?

Description : Returns currently set value.

Response : <NR2> Range 0 to 30

Example : `:PAram:RMSPhase:Upper?`

Example response : `PA:RMSP:U 5.0DEG`

:PAram

Applicable to GSM, DCS and PCS

:RXBERI

:Hirfgenlevel

Description : Sets the RF generator level for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles.

Parameter : <NRf> Range -135 to –15

Allowed suffices : DBM

Example : `:PAram:RXBERI:Hirfgenlevel –104DBM`
Sets the RF generator level for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles to –104 dBm

:RXBERI

:Hirfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range -135 to –15

Example : `:PAram:RXBERI:Hirfgenlevel?`

Example response : `PA:RXBERI:H -104.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXBERI

:Rfgenlevel

Description : Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXBERI:Rfgenlevel –102DBM`
Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles to –102 dBm

:RXBERI

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXBERI:Rfgenlevel?`

Example response : `PA:RXBERI:R –102.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXBERI

:SAmples

Description : Sets the bit sample size for the test.

Parameter : <NR1> Range 1 to 32760000

Allowed suffices : Not applicable

Example : `:PAram:RXBERI:SAmples 45500`
Sets the number of samples to test over to 45500

:RXBERI

:SAmples?

Description : Returns currently set value.

Response : <NR1> Range 1 to 32760000

Example : `:PAram:RXBERI:SAmples?`

Example response : `PA:RXBERI:SA 45500`

:PAram

Applicable to GSM, DCS and PCS

:RXBERI

:STatus

Description : Enables or disables the Class I Bit Error Rate test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXBERI:STatus ON`
Enables the Class I BER test as part of an MI-BASIC test program

:RXBERI

:STatus?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXBERI:STatus?`

Example response : `PA:RXBERI:ST ON`

:PAram

Applicable to GSM, DCS and PCS

:RXBERI

:Upper

Description : Sets the upper % error for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:RXBERI:Upper 0.410PC`
Sets the upper % error for the test to 0.410%

:RXBERI

:Upper?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:RXBERI:Upper?`

Example response : `PA:RXBERI:U 0.410PC`

:PAram

Applicable to GSM, DCS and PCS

:RXBERII

:Hirfgenlevel

Description : Sets the RF generator for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXBERII:Hirfgenlevel –104DBM`
Sets the RF generator level for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles to –104 dBm

:RXBERII

:Hirfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXBERII:Hirfgenlevel?`

Example response : `PA:RXBERII:H –104.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXBERII

:Rfgenlevel

Description : Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXBERII:Rfgenlevel –102DBM`
Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles to –102 dBm

:RXBERII

:Rfgenlevel?

Description : Returns currently set values.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXBERII:Rfgenlevel?`

Example response : `PA:RXBERII:R–102.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXBERII

:SAmples

Description : Sets the bit sample size for the test.

Parameter : <NR1> Range 1 to 14040000

Allowed suffices : Not applicable

Example : `:PAram:RXBERII:SAmples 8200`
Sets the number of samples to test over to 8200

:RXBERII

:SAmples?

Description : Returns currently set value.

Response : <NR1> Range 1 to 14040000

Example : `:PAram:RXBERII:SAmples?`

Example response : `PA:RXBERII:SA 8200`

:PAram

Applicable to GSM, DCS and PCS

:RXBERII

:STatus

Description : Enables or disables the Class II Bit Error Rate test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXBERII:STatus ON`
Enables the Class II BER test as part of an MI-BASIC test program

:RXBERII

:STatus?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXBERII:STatus?`

Example response : `PA:RXBERII:ST ON`

:PAram

Applicable to GSM, DCS and PCS

:RXBERII

:Upper

Description : Sets the upper % error for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:RXBERII:Upper 2.439PC`
Sets the upper % error for the test to 2.439%

:RXBERII

:Upper?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:RXBERII:Upper?`

Example response : `PA:RXBERII:U 2.439PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXDistn

:Averages

Description : Sets the number of averages for the test.
Parameter : <NR1> Range 0 to 99
Allowed suffices : Not applicable
Example : `:PAram:RXDistn:Averages 10`
Sets the number of averages to 10

:RXDistn

:Averages?

Description : Returns currently set value.
Response : <NR1> Range 0 to 99
Example : `:PAram:RXDistn:Averages?`
Example response : `PA:RXD:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXDistn

:Modlevel

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `:PAram:RXDistn:Modlevel 2.3kHz`
Sets the test's reference modulation level to 2.3 kHz

:RXDistn

:Modlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:RXDistn:Modlevel?`
Example response : `PA:RXD:M 2300.0HZ`

:PAram

Applicable to NAMPS and NTACS

:RXDistn

:Nbmodlevel

Description : Sets the reference narrow-band modulation level for the test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:RXDistn:Nbmodlevel 1.5kHz`
Sets the test's narrow-band reference modulation level to 1.5 kHz

:RXDistn

:Nbmodlevel?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:RXDistn:Nbmodlevel?`

Example response : `PA:RXD:N 1500.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXDistn

:RFgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Allowed suffices : `DBM`

Example : `:PAram:RXDistn:RFgenlevel -80DBM`
Sets the RF generator level to –80 dBm for the test

:RXDistn

:RFgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Example : `:PAram:RXDistn:RFgenlevel?`

Example response : `PA:RXD:RF -80.0DBM`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXDistn

:RXfilter

Description : Sets the receiver audio filter type for the Rx Distortion test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:RXDistn:RXfilter CCITT`
Sets the receiver audio filter to CCITT

:RXDistn

:RXfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:RXDistn:RXfilter?`

Example response : `PA:RXD:RX CCITT`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXDistn

:Status

Description : Enables or disables the Rx Distortion test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXDistn:Status ON`
Enables the Rx Distortion test as part of an MI-BASIC test program

:RXDistn

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXDistn:Status?`

Example response : `PA:RXD:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXDistn

:Upper

Description : Sets the % Error Limit for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:RXDistn:Upper 10PC`
Sets the upper error limit to 10%

:RXDistn

:Upper?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:RXDistn:Upper?`

Example response : `PA:RXD:U 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXExpand

:Averages

Description : Sets the number of averages for the test.
Parameter : <NR1> Range 0 to 99
Allowed suffices : Not applicable
Example : `:PAram:RXExpand:Averages 10`
Sets the number of averages to 10

:RXExpand

:Averages?

Description : Returns currently set value.
Response : <NR1> Range 0 to 99
Example : `:PAram:RXExpand:Averages?`
Example response : `PA:RXE:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXExpand

:Error

Description : Sets the % error limit for the test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : `PC`
Example : `:PAram:RXExpand:Error 10PC`
Sets the test error limit to 10%

:RXExpand

:Error?

Description : Returns currently set value.
Response : <NR2> Range 0 to 1000
Example : `:PAram:RXExpand:Error?`
Example response : `PA:RXE:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXExpand

:Modlevel

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `:PAram:RXExpand:Modlevel 2.3kHz`
Sets the test's reference modulation level to 2.3 kHz

:RXExpand

:Modlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:RXExpand:Modlevel?`
Example response : `PA:RXE:M 2300.0HZ`

:PAram

Applicable to NAMPS and NTACS

:RXExpand

:Nbmodlevel

Description : Sets the reference narrow-band modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `:PAram:RXExpand:Nbmodlevel 1.5kHz`
Sets the test's narrow-band reference modulation level to 1.5 kHz

:RXExpand

:Nbmodlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:RXExpand:Nbmodlevel?`
Example response : `PA:RXE:N 1500.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXExpand

:REf

Description : Sets the number of averages for the test.

Parameter : <NR1> Range 0 to 99

Allowed suffices : Not applicable

Example : `:PAram:RXExpand:REf 2`
Sets the expansion test reference to be a factor of 2

:RXExpand

:REf?

Description : Returns currently set value.

Response : <NR1> Range 0 to 99

Example : `:PAram:RXExpand:REf?`

Example response : `PA:RXE:RE 2`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXExpand

:RXfilter

Description : Sets the receiver audio filter type for the Rx Expansion test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:RXExpand:RXfilter CCITT`
Sets the receiver audio filter to CCITT

:RXExpand

:RXfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:RXExpand:RXfilter?`

Example response : `PA:RXE:RX CCITT`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXExpand

:STatus

Description : Enables or disables the Rx Expansion test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXExpand:STatus ON`
Enables the Rx Expansion test as part of an MI-BASIC test program

:RXExpand

:STatus?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXExpand:STatus?`

Example response : `PA:RXE:ST ON`

:PAram

Applicable to PMR

:RXExpand

:SWing

Description : Sets the test's swing range in dB. For example, if a value of 10 dB is specified the compression test will 'swing' the modulation deviation between –5 dB and +5 dB of the reference modulation level.

Parameter : <NRf> Range 0 to 100

Allowed suffices : `DB`

Example : `:PAram:RXExpand:SWing 10DB`
Sets the test swing range to 10 dB

:RXExpand

:SWing?

Description : Returns currently set value.

Response : <NR2> Range 0 to 100

Example : `:PAram:RXExpand:SWing?`

Example response : `PA:RXE:SW 10.0DB`

:PAram

Applicable to GSM, DCS and PCS

:RXFer

:Hirgfenlevel

Description : Sets the RF generator level for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXFer:Hirfgenlevel –104DBM`
Sets the RF generator level for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles to –104 dBm

:RXFer

:Hirfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXFer:Hirfgenlevel?`

Example response : `PA:RXF:H –104.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXFer

:Rfgenlevel

Description : Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXFer:Rfgenlevel –102DBM`
Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles to –102 dBm

:RXFer

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXFer:Rfgenlevel?`

Example response : `PA:RXF:R –102.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXFer

:SAmples

Description : Sets the bit sample size for the test.

Parameter : <NR1> Range 1 to 180000

Allowed suffices : Not applicable

Example : `:PAram:RXFer:SAmples 500`
Sets the number of samples to test over to 500

:RXFer

:SAmples?

Description : Returns currently set value.

Response : <NR1> Range 1 to 180000

Example : `:PAram:RXFer:SAmples?`

Example response : `PA:RXF:SA 500`

:PAram

Applicable to GSM, DCS and PCS

:RXFer

:STatus

Description : Enables or disables the Frame Erasure Rate test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXFer:STatus ON`
Enables the Frame Erasure Rate test as part of an MI-BASIC test program

:RXFer

:STatus?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXFer:STatus?`

Example response : `PA:RXF:ST ON`

:PAram

Applicable to GSM, DCS and PCS

:RXFer

:Upper

Description : Sets the upper % error limit for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:RXFer:Upper 0.2PC`
Sets the upper % error limit for the test to 0.2%

:RXFer

:Upper?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:RXFer:Upper?`

Example response : `PA:RXF:U 0.200PC`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERIB

:Hirfgenlevel

Description : Sets the RF generator for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXRBERIB:Hirfgenlevel –104DBM`
Sets the RF generator level for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles to –104 dBm

:RXRBERIB

:Hirfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXRBERIB:Hirfgenlevel?`

Example response : `PA:RXRBERIB:H -104.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERIB

:Rfgenlevel

Description : Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXRBERIB:Rfgenlevel -102DBM`
Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles to –102 dBm

:RXRBERIB

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXRBERIB:Rfgenlevel?`

Example response : `PA:RXRBERIB:R -102.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERIB

:SAmples

Description : Sets the bit sample size for the test.

Parameter : <NR1> Range 1 to 23760000

Allowed suffices : Not applicable

Example : `:PAram:RXRBERIB:SAmples 33000`
Sets the number of samples to test over to 33000

:RXRBERIB

:SAmples?

Description : Returns currently set value.

Response : <NR1> Range 1 to 23760000

Example : `:PAram:RXRBERIB:SAmples?`

Example response : `PA:RXRBERIB:SA 33000`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERIB

:STatus

Description : Enables or disables the Class Ib Residual Bit Error Rate test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXRBERIB:STatus ON`
Enables the Class Ib RBER test as part of an MI-BASIC test program

:RXRBERIB

:STatus?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXRBERIB:STatus?`

Example response : `PA:RXRBERIB:ST ON`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERIB

:Upper

Description : Sets the upper % error for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:RXRBERIB:Upper 0.410PC`
Sets the upper % error for the test to 0.410%

:RXRBERIB

:Upper?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:RXRBERIB:Upper?`

Example response : `PA:RXRBERIB:U 0.410PC`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERII

:Hirfgenlevel

Description : Sets the RF generator level for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXRBERII:Hirfgenlevel –104DBM`
Sets the RF generator level for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles to –104 dBm

:RXRBERII

:Hirfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXRBERII:Hirfgenlevel?`

Example response : `PA:RXRBERII:H –104.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERII

:Rfgenlevel

Description : Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXRBERII:Rfgenlevel –102DBM`
Sets the RF generator level for GSM Class 4 and 5 and DCS/PCS Class 1 and 2 mobiles to –102 dBm

:RXRBERII

:Rfgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXRBERII:Rfgenlevel?`

Example response : `PA:RXRBERII:R –102.0DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERII

:SAmples

Description : Sets the bit samples size for the test.

Parameter : <NR1> Range 1 to 14040000

Allowed suffices : Not applicable

Example : `:PAram:RXRBERII:SAmples 8200`
Sets the number of samples to test over to 8200

:RXRBERII

:SAmples?

Description : Returns currently set value.

Response : <NR1> Range 1 to 14040000

Example : `:PAram:RXRBERII:SAmples?`

Example response : `PA:RXRBERII:SA 8200`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERII

:STatus

Description : Enables or disables the Class II Residual Bit Error Rate test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXRBERII:STatus ON`
Enables the Class II RBER test as part of an MI-BASIC test program

:RXRBERII

:STatus?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXRBERII:STatus?`

Example response : `PA:RXRBERII:ST ON`

:PAram

Applicable to GSM, DCS and PCS

:RXRBERII

:Upper

Description : Sets the upper % error for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:RXRBERII:Upper 2.439PC`
Sets the upper % error for the test to 2.439%

:RXRBERII

:Upper?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:RXRBERII:Upper?`

Example response : `PA:RXRBERII:U 2.439PC`

:PAram

Applicable to GSM, DCS and PCS

:RXRSsi

:Decrement

Description : Sets the value by which to decrement the RF generator level for the test as part of an MI-BASIC test program.

Parameter : <NRf> Range 1 to 90

Allowed suffices : DB

Example : `:PAram:RXRSsi:Decrement 12DB`
Sets the RF generator level decrement value to 12 dB for an MI-BASIC test program.

:RXRSsi

:Decrement?

Description : Returns currently set value.

Response : <NR2> Range 1 to 90

Example : `:PAram:RXRSsi:Decrement?`

Example response : `PA:RXRS:D 12.0DB`

:PAram

Applicable to GSM, DCS and PCS

:RXRSsi

:Error

Description : Sets the error tolerance (for RF generator levels of less than –70 dBm) for the test.

Parameter : <NRf> Range 0 to 100

Allowed suffices : DB

Example : `:PAram:RXRSsi:Error 4DB`
Sets the error tolerance to 4 dB for the test

:RXRSsi

:Error?

Description : Returns currently set value.

Response : <NR2> Range 0 to 100

Example : `:PAram:RXRSsi:Error?`

Example response : `PA:RXRS:E 4.0DB`

:PAram

Applicable to GSM, DCS and PCS

:RXRSsi

:Firstvalue

Description : Sets the first RF generator level for the test as part of an MI- BASIC test program.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXRSsi:Firstvalue -48.5DBM`
Sets the first RF generator level to –48.5 dBm for an MI- BASIC test program

:RXRSsi

:Firstvalue?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXRSsi:Firstvalue?`

Example response : `PA:RXRS:F –48.5DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXRSsi

:Hierror

Description : Sets the error tolerance (for RF generator levels of greater than –70 dBm) for the test.

Parameter : <NRf> Range 0 to 100

Allowed suffices : DB

Example : `:PAram:RXRSsi:Hierror 6DB`
Sets the error tolerance to 6 dB for the test

:RXRSsi

:Hierror?

Description : Returns currently set value.

Response : <NR2> Range 0 to 100

Example : `:PAram:RXRSsi:Hierror?`

Example response : `PA:RXRS:H 6.0DB`

:PAram

Applicable to GSM, DCS and PCS

:RXRSsi

:Lastvalue

Description : Sets the last RF generator level for the test as part of an MI-BASIC test program.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXRSsi:Lastvalue -109.5DBM`
Sets the last RF generator level to –109.5 dBm for an MI-BASIC test program

:RXRSsi

:Lastvalue?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXRSsi:Lastvalue?`

Example response : `PA:RXRS:L -109.5DBM`

:PAram

Applicable to GSM, DCS and PCS

:RXRSsi

:Status

Description : Enables or disables the Received Signal Strength Indication test as part of an MI-BASIC test program.

Parameter : <CDP> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXRSsi:Status ON`
Enables the RSSI test as part of an MI-BASIC test program

:RXRSsi

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXRSsi:Status?`

Example response : `PA:RXRS:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSEns

:Averages

Description : Sets the number of averages for the test.

Parameter : <NR1> Range 0 to 99

Allowed suffices : Not applicable

Example : `:PAram:RXSEns:Averages 10`
Sets the number of averages to 10

:RXSEns

:Averages?

Description : Returns currently set value.

Response : <NR1> Range 0 to 99

Example : `:PAram:RXSEns:Averages?`

Example response : `PA:RXSE:A 10`

:PAram

Applicable to GSM, DCS and PCS

:RXSEns

:Hiupper

Description : Sets the upper RF generator level limit for GSM Class 1 to 3 and DCS/PCS Class 3 mobiles.

Parameter : <NRf> Range –135 to –15

Allowed suffices : DBM

Example : `:PAram:RXSEns:Hiupper –104DBM`
Sets the GSM Class 1-3 and DCS/PCS Class 3 RF generator level upper limit to –104 dBm

:RXSEns

:Hiupper?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example : `:PAram:RXSEns:Hiupper?`

Example response : `PA:RXSE:H –104.0DBM`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSEns

:Modlevel

Description : Sets the reference modulation level for the test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : HZ, KHZ, MHZ

Example : `:PAram:RXSEns:Modlevel 2.3kHz`
Sets the test's reference modulation level to 2.3 kHz

:RXSEns

:Modlevel?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:RXSEns:Modlevel?`

Example response : `PA:RXSE:M 2300.0HZ`

:PAram

Applicable to NAMPS and NTACS

:RXSEns

:Nbmodlevel

Description : Sets the reference narrow-band modulation level for the test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:RXSEns:Nbmodlevel 1.5kHz`

Sets the test's narrow-band reference modulation level to 1.5 kHz

:RXSEns

:Nbmodlevel?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:RXSEns:Nbmodlevel?`

Example response : `PA:RXSE:N 1500.0HZ`

:PAram

Applicable to GSM, DCS and PCS

:RXSEns

:REFBer

Description : Sets the Reference Residual Bit Error Rate (RBER) for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : `PC`

Example : `:PAram:RXSEns:REFBer 2.439PC`

Sets the Reference RBER to 2.439% for the test

:RXSEns

:REFBer?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:RXSEns:REFBer?`

Example response : `PA:RXSE:REFB 2.439PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSEns

:REFSinad

Description : Sets the Reference SINAD level for the test.

Parameter : <NRf> Range 0 to 100

Allowed suffices : DB

Example : `:PAram:RXSEns:REFSinad 12dB`
Sets the test SINAD reference to 12 dB

:RXSEns

:REFSinad?

Description : Returns currently set value.

Response : <NR2> Range 0 to 100

Example : `:PAram:RXSEns:REFSinad?`

Example response : `PA:RXSE:REFS 12.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSEns

:RXfilter

Description : Sets the receiver audio filter type for the Rx Sensitivity test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:RXSEns:RXfilter CCITT`
Sets the receiver audio filter type to CCITT

:RXSEns

:RXfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:RXSEns:RXfilter?`

Example response : `PA:RXSE:RX CCITT`

:PAram

Applicable to GSM, DCS and PCS

:RXSEns

:SAmples

Description : Sets the number of samples over which to measure the Bit Error Rate.

Parameter : <NR1> Range 1 to 14040000

Allowed suffices : Not applicable

Example : `:PAram:RXSEns:SAmples 8200`
Sets the number of samples to measure over to 8200

:RXSEns

:SAmples?

Description : Returns currently set value.

Response : <NR1> Range 1 to 14040000

Example : `:PAram:RXSEns:SAmples?`

Example response : `PA:RXSE:SA 8200`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:RXSEns

:STatus

Description : Enables or disables the Rx Sensitivity test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXSEns:STatus ON`
Enables the Rx Sensitivity test as part of an MI-BASIC test program

:RXSEns

:STatus?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXSEns:STatus?`

Example response : `PA:RXSE:ST ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:RXSEns

:Upper

Description : Sets the RF generator level upper limit for the test.
Note: For GSM, this is for Class 4 and 5 mobiles only. For DCS or PCS, this is for Class 1 and 2 mobiles only.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Allowed suffices : DBM

Example : `:PAram:RXSEns:Upper -116DBM`
Sets the RF generator level upper limit to –116 dBm for the test

:RXSEns

:Upper?

Description : Returns currently set value.

Response : <NR2> Range -135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)
Range –135 to –15 (GSM, DCS or PCS)

Example : `:PAram:RXSEns:Upper?`

Example response : `PA:RXSE:U -116.0DBM`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSInad

:Averages

Description : Sets the number of averages for the test.

Parameter : <NR1> Range 0 to 99

Allowed suffices : Not applicable

Example : `:PAram:RXSInad:Averages 10`
Sets the number of averages to 10

:RXSInad

:Averages?

Description : Returns currently set value.

Response : <NR1> Range 0 to 99

Example : `:PAram:RXSInad:Averages?`

Example response : `PA:RXSI:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSInad

:Lower

Description : Sets the lower pass limit for the test.
Parameter : <NRf> Range 0 to 100
Allowed suffices : DB
Example : `:PAram:RXSInad:Lower 12DB`
Sets the % lower pass limit to 12 dB

:RXSInad

:Lower?

Description : Returns currently set value.
Response : <NR2> Range 0 to 100
Example : `:PAram:RXSInad:Lower?`
Example response : `PA:RXSI:L 12.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSInad

:Modlevel

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : `:PAram:RXSInad:Modlevel 2.3kHz`
Sets the test's reference modulation level to 2.3 kHz

:RXSInad

:Modlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:RXSInad:Modlevel?`
Example response : `PA:RXSI:M 2300.0HZ`

:PAram

Applicable to NAMPS and NTACS

:RXSInad

:Nbmodlevel

Description : Sets the reference narrow-band modulation level for the test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:RXSInad:Nbmodlevel 1.5kHz`

Sets the test's narrow-band reference modulation level to 1.5 kHz

:RXSInad

:Nbmodlevel?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:RXSInad:Nbmodlevel?`

Example response : `PA:RXSI:N 1500.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSInad

:RFgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Allowed suffices : `DBM`

Example : `:PAram:RXSInad:RFgenlevel -80DBM`

Sets the RF generator level to –80 dBm for the test

:RXSInad

:RFgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Example : `:PAram:RXSInad:RFgenlevel?`

Example response : `PA:RXSI:RF -80.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSInad

:RXfilter

Description : Sets the receiver audio filter type for the Rx Distortion test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:RXSInad:RXfilter CCITT`
Sets the receiver audio filter to CCITT

:RXSInad

:RXfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:RXSInad:RXfilter?`

Example response : `PA:RXSI:RX CCITT`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSInad

:Status

Description : Enables or disables the Rx SINAD test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXSInad:Status ON`
Enables the Rx SINAD test as part of an MI-BASIC test program.

:RXSInad

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXSInad:Status?`

Example response : `PA:RXSI:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSN

:Averages

Description : Sets the number of averages for the test.

Parameter : <NR1> Range 0 to 99

Allowed suffices : Not applicable

Example : `:PAram:RXSN:Averages 10`
Sets the number of averages to 10

:RXSN

:Averages?

Description : Returns currently set value.

Response : <NR1> Range 0 to 99

Example : `:PAram:RXSN:Averages?`

Example response : `PA:RXSN:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSN

:Lower

Description : Sets the lower pass limit for the test.
Parameter : <NRf> Range 0 to 100
Allowed suffices : DB
Example : `:PAram:RXSN:Lower 10DB`
Sets the % lower pass limit to 10 dB

:RXSN

:Lower?

Description : Returns currently set value.
Response : <NR2> Range 0 to 100
Example : `:PAram:RXSN:Lower?`
Example response : `PA:RXSN:L 10.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSN

:Modlevel

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : `:PAram:RXSN:Modlevel 2.3kHz`
Sets the test's reference modulation level to 2.3 kHz

:RXSN

:Modlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:RXSN:Modlevel?`
Example response : `PA:RXSN:M 2300.0HZ`

:PAram

Applicable to NAMPS and NTACS

:RXSN

:Nbmodlevel

Description : Sets the reference narrow-band modulation level for the test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:RXSN:Nbmodlevel 1.5kHz`
Sets the test's narrow-band reference modulation level to 1.5 kHz

:RXSN

:Nbmodlevel?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:RXSN:Nbmodlevel?`

Example response : `PA:RXSN:N 1500.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSN

:RFgenlevel

Description : Sets the RF generator level for the test.

Parameter : <NRf> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Allowed suffices : `DBM`

Example : `:PAram:RXSN:RFgenlevel -80DBM`
Sets the RF generator level to –80 dBm for the test

:RXSN

:RFgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to +13 (2965A)
Range -135 to +10 (2966A, 2967)

Example : `:PAram:RXSN:RFgenlevel?`

Example response : `RA:RXSN:RF -80.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSN

:RXfilter

Description : Sets the receiver audio filter type for the Rx S/N test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:RXSN:RXfilter CCITT`
Sets the receiver audio filter to CCITT

:RXSN

:RXfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:RXSN:RXfilter?`

Example response : `PA:RXSN:RX CCITT`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:RXSN

:Status

Description : Enables or disables the Rx S/N test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:RXSN:Status ON`
Enables the Rx S/N test as part of an MI-BASIC test program

:RXSN

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:RXSN:Status?`

Example response : `PA:RXSN:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and NMT

:SATDevn

:Error

Description : Sets the % error limit for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:SATDevn:Error 10PC`
Sets the test error limit to 10%

:SATDevn

:Error?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:SATDevn:Error?`

Example response : `PA:SATD:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and NMT

:SATDevn

:Ref

Description : Sets the reference for the SAT deviation test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : `:PAram:SATDevn:Ref 1.7kHz`
Sets the test's reference deviation level to 1.7 kHz

:SATDevn

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:SATDevn:Ref?`
Example response : `PA:SATD:R 1700.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and NMT

:SATDevn

:Status

Description : Enables or disables the SAT Deviation test as part of an MI-BASIC test program.
Parameter : <CPD> OFF, ON
Allowed suffices : Not applicable
Example : `:PAram:SATDevn:Status ON`
Enables the SAT deviation test as part of an MI-BASIC test program

:SATDevn

:Status?

Description : Returns currently set value.
Response : <CRD> OFF, ON
Example : `:PAram:SATDevn:Status?`
Example response : `PA:SATD:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and NMT

:SATFreq

:Error

Description : Sets the % error limit for the test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : PC
Example : `:PAram:SATFreq:Error 10PC`
Sets the test error limit to 10%

:SATFreq

:Error?

Description : Returns currently set value.
Response : <NR2> Range 0 to 1000
Example : `:PAram:SATFreq:Error?`
Example response : `PA:SATF:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and NMT

:SATFreq

:Ref

Description : Sets the reference frequency for the test.
Parameter : <NRf> Range 0 to 20 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : `:PAram:SATFreq:Ref 6kHz`
Sets the test's reference SAT frequency to 6 kHz

:SATFreq

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0 to 20 kHz
Example : `:PAram:SATFreq:Ref?`
Example response : `PA:SATF:R 6000.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS and NMT

:SATFreq

:Status

Description : Enables or disables the SAT Frequency test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example :
```
:PAram:SATFreq:Status ON
```
Enables the SAT deviation test as part of an MI-BASIC test program

:SATFreq

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example :
```
:PAram:SATFreq:Status?
```

Example response :
```
PA:SATF:S ON
```

:PAram

Applicable to GSM, DCS and PCS

:SPEechqual

:Rfgenlevel

Description : Sets the RF generator level for the Speech Quality test.

Parameter : <NRf> Range –135 to –15

Allowed suffices :
```
DBM
```

Example :
```
:PAram:SPEechqual:Rfgenlevel –80.0DBM
```
Sets the RF generator level to –80 dBm for the test

:SPEechqual

:RFgenlevel?

Description : Returns currently set value.

Response : <NR2> Range –135 to –15

Example :
```
:PAram:SPEechqual:Rfgenlevel?
```

Example response :
```
PA:SPE:R –80.0DBM
```

:PAram

Applicable to GSM, DCS and PCS

:SPEechqual

:Status

Description : Enables or disables the Speech Quality test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:SPEechqual:Status ON`
Enables the Speech Quality test as part of an MI-BASIC test program

:SPEechqual

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:SPEechqual:Status?`

Example response : `PA:SPE:S ON`

:PAram

Applicable to GSM, DCS and PCS

:SPEechqual

:Timeout

Description : Sets the response timeout limit for the test.

Parameter : <NR2> Range 0.001 to 600

Allowed suffices : `S`

Example : `:PAram:SPEechqual:Timeout 20S`
Sets the test's timeout to 20 seconds for a response to be received

:SPEechqual

:Timeout?

Description : Returns currently set value.

Response : <NR2> Range 0.001 to 600

Example : `:PAram:SPEechqual:Timeout?`

Example response : `PA:SPE:T 20.0S`

:PAram

Applicable to PMR

:SPOtfreq

:Rxchann *n*,

Description : Allocates a receive channel frequency to a receive channel number specified by the number *n*.

Parameters : <NR1> Range *n* = 1 to 16
<NRf> Range 100 kHz to 1000 MHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:SPOtfreq:Rxchann 5,50MHZ`
Sets the frequency of receive channel number 5 to 50 MHz

:SPOtfreq

:Rxchann? *n*

Description : Returns currently set frequency for the receive channel specified by the number *n*.

Response : <NR2> Range 100,000 to 1000,000,000 Hz

Example : `:PAram:SPOtfreq:Rxchann? 5`

Example response : `PA:SPO:R 50000000.0HZ`

:PAram

Applicable to PMR

:SPOtfreq

:Status

Description : Enables or disables Spot Frequency test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:SPOtfreq:Status ON`
Enables the Spot Frequencies to be used as part of an MI-BASIC test program

:SPOtfreq

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:SPOtfreq:Status?`

Example response : `PA:SPO:S ON`

:PAram

Applicable to PMR

:SPOtfreq

:Txchann *n*,

Description : Allocates a transmit channel frequency to a transmit channel number specified by the number *n*.

Parameters : <NR1> *n* = Range 1 to 16
<NRf> Range 100 kHz to 1000 MHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:SPOtfreq:Txchann 5,50MHz`
Sets the frequency of transmit channel number 5 to 50 MHz

:SPOtfreq

:Txchann? *n*

Description : Returns currently set frequency for the transmit channel specified by the number *n*.

Response : <NR2> Range 100,000 to 1000,000,000 Hz

Example : `:PAram:SPOtfreq:Txchann? 5`

Example response : `PA:SPO:T 50000000.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:STDEvn

:Error

Description : Sets the % error limit for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : `PC`

Example : `:PAram:STDEvn:Error 10PC`
Sets the test error limit to 10%

:STDEvn

:Error?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:STDEvn:Error?`

Example response : `PA:STDE:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:STDEvn

:Ref

Description : Sets the reference for the ST deviation test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : `:PAram:STDEvn:Ref 6.4kHz`
Sets the test's reference deviation level to 6.4 kHz

:STDEvn

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:STDEvn:Ref?`
Example response : `PA:STDE:R 6400.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:STDEvn

:Status

Description : Enables or disables the ST Deviation test as part of an MI-BASIC test program.
Parameter : <CPD> OFF, ON
Allowed suffices : Not applicable
Example : `:PAram:STDEvn:Status ON`
Enables the ST deviation test as part of an MI-BASIC test program

:STDEvn

:Status?

Description : Returns currently set value.
Response : <CRD> OFF, ON
Example : `:PAram:STDEvn:Status?`
Example response : `PA:STDE:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:STDUrn

:Error

Description : Sets the % error limit for the test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : PC
Example : `:PAram:STDUrn:Error 10PC`
Sets the test error limit to 10%

:STDUrn

:Error?

Description : Returns currently set value.
Response : <NR2> Range 0 to 1000
Example : `:PAram:STDUrn:Error?`
Example response : `PA:STDU:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:STDUrn

:Ref

Description : Sets the response timeout limit for the test.
Parameter : <NR2> Range 0.001 to 600
Allowed suffices : S
Example : `:PAram:STDUrn:Ref 20S`
Sets the reference to 20 seconds

:STDUrn

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0.001 to 600
Example : `:PAram:STDUrn:Ref?`
Example response : `PA:STDU:R 20.0S`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:STDUrn

:Status

Description : Enables or disables the ST Duration test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:STDUrn:Status ON`
Enables the ST Duration test as part of an MI-BASIC test program

:STDUrn

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:STDUrn:Status?`

Example response : `PA:STDU:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:STFreq

:Error

Description : Sets the % error limit for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:STFreq:Error 10PC`
Sets the test error limit to 10%

:STFreq

:Error?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:STFreq:Error?`

Example response : `PA:STF:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:STFreq

:Ref

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 20 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : :PAram:STFreq:Ref 8kHz
Sets the test's reference ST frequency to 8 kHz

:STFreq

:Ref?

Description : Returns currently set value.
Response : <NR2> Range 0 to 20 kHz
Example : :PAram:STFreq:Ref?
Example response : PA:STF:R 8000.0HZ

:PAram

Applicable to AMPS, NAMPS, TACS and NTACS

:STFreq

:Status

Description : Enables or disables the ST Frequency test as part of an MI-BASIC test program.
Parameter : <CPD> OFF, ON
Allowed suffices : Not applicable
Example : :PAram:STFreq:Status ON
Enables the ST Frequency test as part of an MI-BASIC test program

:STFreq

:Status?

Description : Returns currently set value.
Response : <CRD> OFF, ON
Example : :PAram:STFreq:Status?
Example response : PA:STF:S ON

:PAram

Applicable to GSM, DCS and PCS

:TIming

:Error

Description : Sets the error tolerance for the Tx Timing test.

Parameter : <NRf> Range 0 to 100

Allowed suffices : BITS

Example : `:PAram:TIming:Error 1.00BITS`
Sets the error tolerance to 1 bit for the test

:TIming

:Error?

Description : Returns currently set value.

Response : <NR2> Range 0 to 100

Example : `:PAram:TIming:Error?`

Example response : `PA:TI:E 1.00BITS`

:PAram

Applicable to GSM, DCS and PCS

:TIming

:Firstvalue

Description : Sets the first timing advance value for the Tx Timing test as part of an MI-BASIC test program.

Parameter : <NRf> Range 0 to 63

Allowed suffices : BITS

Example : `:PAram:TIming:Firstvalue 0BITS`
Sets the first timing advance value to 0 bits for an MI-BASIC test program

:TIming

:Firstvalue?

Description : Returns currently set value.

Response : <NR2> Range 0 to 63

Example : `:PAram:TIming:Firstvalue?`

Example response : `PA:TI:F 0.0BITS`

:PAram

Applicable to GSM, DCS and PCS

:TIming

:Increment

Description : Sets the value by which to increment the timing advance for the Tx Timing test as part of an MI-BASIC test program.

Parameter : <NRf> Range 0 to 63

Allowed suffices : BITS

Example : `:PAram:TIming:Increment 10BITS`
Sets the timing advance increment to 10 bits for an MI-BASIC test program.

:TIming

:Increment?

Description : Returns currently set value.

Response : <NR2> Range 0 to 63

Example : `:PAram:TIming:Increment?`

Example response : `PA:TI:I 10.0BITS`

:PAram

Applicable to GSM, DCS and PCS

:TIming

:Lastvalue

Description : Sets the last timing advance value for the Tx Timing test as part of an MI-BASIC test program.

Parameter : <NRf> Range 0 to 63

Allowed suffices : BITS

Example : `:PAram:TIming:Lastvalue 63BITS`
Sets the last timing advance value to 63 bits for aMI-BASIC test program

:TIming

:Lastvalue?

Description : Returns currently set value.

Response : <NR2> Range 0 to 63

Example : `:PAram:TIming:Lastvalue?`

Example response : `PA:TI:L 63.0BITS`

:PAram

Applicable to GSM, DCS and PCS

:TIming

:Status

Description : Enables or disables the Tx Timing test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TIming:Status ON`
Enables the Tx Timing test as part of an MI-BASIC test program

:TIming

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TIming:Status?`

Example response : `PA:TI:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXCompress

:Averages

Description : Sets the number of averages for the Tx Compression test.

Parameter : <NR1> Range 0 to 99

Allowed suffices : Not applicable

Example : `:PAram:TXCompress:Averages 10`
Sets the number of averages to 10

:TXCompress

:Averages?

Description : Returns currently set value.

Response : <NR1> Range 0 to 99

Example : `:PAram:TXCompress:Averages?`

Example response : `PA:TXC:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXCompress

:Error

Description : Sets the % error limit for the Tx Compression test.
Parameter : <NRf> Range 0 to 1000
Allowed suffices : Not applicable
Example : `:PAram:TXCompress:Error 10PC`
Sets the test error limit to 10%

:TXCompress

:Error?

Description : Returns currently set value.
Response : <NR2> Range 0 to 1000
Example : `:PAram:TXCompress:Error?`
Example response : `PA:TXC:E 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXCompress

:Modlevel

Description : Sets the reference modulation level for the Tx Compression test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `:PAram:TXCompress:Modlevel 6kHz`
Sets the test's reference modulation level to 6 kHz

:TXCompress

:Modlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:TXCompress:Modlevel?`
Example response : `PA:TXC:M 6000.0HZ`

:PAram

Applicable to NAMPS and NTACS

:TXCompress

:Nbmodlevel

Description : Sets the reference narrow-band modulation level for the Tx Compression test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:TXCompress:Nbmodlevel 1.5kHz`
Sets the test's narrow-band reference modulation level to 1.5 kHz

:TXCompress

:Nbmodlevel?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:TXCompress:Nbmodlevel?`

Example response : `PA:TXC:N 1500.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXCompress

:Ref

Description : Sets the number of averages for the Tx Compression test.

Parameter : <NR1> Range 0 to 99

Allowed suffices : Not applicable

Example : `:PAram:TXCompress:Ref 2`
Sets the compression test reference to a factor of 2

:TXCompress

:Ref?

Description : Returns currently set value.

Response : <NR1> Range 0 to 99

Example : `:PAram:TXCompress:Ref?`

Example response : `PA:TXC:R 2`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXCompress

:STatus

Description : Enables or disables the Tx Compression test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TXCompress:STatus ON`
Enables the Tx Compression test as part of an MI-BASIC test program.

:TXCompress

:STatus?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TXCompress:STatus?`

Example response : `PA:TXC:ST ON`

:PAram

Applicable to PMR

:TXCompress

:SWing

Description : Sets the Tx Compression test's swing range in dB. For example, if a value of 10 dB is specified, the compression test will "swing" the modulation deviation between –5 dB and +5 dB of the reference modulation level.

Parameter : <NRf> Range 0 to 100

Allowed suffices : `DB`

Example : `:PAram:TXCompress:SWing 10DB`
Sets the test swing range to 10 dB

:TXCompress

:SWing?

Description : Returns currently set value.

Response : <NR2> Range 0 to 100

Example : `:PAram:TXCompress:SWing?`

Example response : `PA:TXC:SW 10.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXCompress

:Txfilter

Description : Sets the demodulation filter type for the Tx Compression test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:TXCompress:Txfilter CCITT`
Sets the demodulation filter to CCITT

:TXCompress

:Txfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:TXCompress:Txfilter?`

Example response : `PA:TXC:T CCITT`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXDistn

:Averages

Description : Sets the number of averages for the Tx Distortion test.
Parameter : <NR1> Range 0 to 99
Allowed suffices : Not applicable
Example : `:PAram:TXDistn:Averages 10`
Sets the number of averages to 10

:TXDistn

:Averages?

Description : Returns currently set value.
Response : <NR1> Range 0 to 99
Example : `:PAram:TXDistn:Averages?`
Example response : `PA:TXD:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXDistn

:Modlevel

Description : Sets the reference modulation level for the Tx Distortion test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `:PAram:TXDistn:Modlevel 2.3kHz`
Sets the test's reference modulation level to 2.3 kHz

:TXDistn

:Modlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:TXDistn:Modlevel?`
Example response : `PA:TXD:M 2300.0HZ`

:PAram

Applicable to NAMPS and NTACS.

:TXDistn

:Nbmodlevel

Description : Sets the reference narrow-band modulation level for the Tx Distortion test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:TXDistn:Nbmodlevel 1.5kHz`
Sets the test's narrow-band reference modulation level to 1.5 kHz

:TXDistn

:Nbmodlevel?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:TXDistn:Nbmodlevel?`

Example response : `PA:TXD:N 1500.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXDistn

:Status

Description : Enables or disables the Tx Distortion test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TXDistn:Status ON`
Enables the Tx Distortion test as part of an MI-BASIC test program.

:TXDistn

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TXDistn:Status?`

Example response : `PA:TXD:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXDistn

:Txfilter

Description : Sets the demodulation filter type for the Tx Distortion test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:TXDistn:Txfilter CCITT`
Sets the demodulation filter to CCITT

:TXDistn

:Txfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:TXDistn:Txfilter?`

Example response : `PA:TXD:T CCITT`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXDistn

:Upper

Description : Sets the upper error limit for the Tx Distortion test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:TXDistn:Upper 10PC`
Sets the upper error limit to 10%

:TXDistn

:Upper?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:TXDistn:Upper?`

Example response : `PA:TXD:U 10.0PC`

:PAram

Applicable to GSM, DCS and PCS

:TXFreq

:Bursttst

Description : Sets the number of bursts tested for the test.

Parameter : <NR1> Range 1 to 100

Allowed suffices : Not applicable

Example : `:PAram:TXFreq:Bursttst 20`
Sets the number of bursts to test to 20

:TXFreq

:Bursttst?

Description : Returns currently set value.

Response : <NR1> Range 1 to 100

Example : `:PAram:TXFreq:Bursttst?`

Example response : `PA:TXF:B 20`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:TXFreq

:Error

Description : Sets the frequency error limit for the test.

Parameter : <NRf> Range 0 to 1000% (all systems except GSM, DCS or PCS)
Range 1 to 1100 MHz (GSM, DCS or PCS)
Range 1 to 10^9 ppm (GSM, DCS or PCS)

Allowed suffices : `PC` (all systems except GSM, DCS or PCS)
`HZ, KHZ, MHz, PPM` (GSM, DCS or PCS)

Examples : `:PAram:TXFreq:Error 10PC`
Sets the test error limit to 10%

`:PAram:TXFreq:Error 90HZ`
Sets the test error limit to 90 Hz

:TXFreq

:Error?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000 (all systems except GSM, DCS or PCS)
Range 1 to 1100 MHz (GSM, DCS or PCS)
Range 1 to 10^9 ppm (GSM, DCS or PCS)

Example : `:PAram:TXFreq:Error?`

Example response : `PA:TXF:E 10.0PC`

:PAram

Applicable to GSM, DCS and PCS

:TXFreq

:Meastype

Description : Defines the type of measurement for the test.

Parameter : <CPD> WORSTCASE
AVERAGE

Allowed suffices : Not applicable

Example : `:PAram:TXFreq:Meastype WORSTCASE`
Sets the test's measurement type to worst case

:TXFreq

:Meastype?

Description : Returns currently set value for the test

Parameter : <CRD> WORSTCASE
AVERAGE

Allowed suffices : Not applicable

Example : `:PAram:TXFreq:Meastype?`

Example response : `:PA:TXF:M WORSTCASE`

:PAram

Applicable to NAMPS and NTACS

:TXFreq

:Nberror

Description : Sets the narrow-band error limit for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : PC

Example : `:PAram:TXFreq:Nberror 10PC`
Sets the test's narrow-band error limit to 10%

:TXFreq

:Nberror?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:TXFreq:Nberror?`

Example response : `PA:TXF:N 10.0PC`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:TXFreq

:Ref

Description : Sets the reference frequency for the test.

Parameter : <NRf> Range 100 kHz to 1 GHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:TXFreq:Ref 800MHz`
Sets the test's reference transmitter frequency to 800 MHz

:TXFreq

:Ref?

Description : Returns currently set value.

Response : <NR2> 100,000 to 1000,000,000 Hz

Example : `:PAram:TXFreq:Ref?`

Example response : `PA:TXF:R 800000000.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:TXFreq

:Status

Description : Enables or disables the Tx Frequency test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TXFreq:Status ON`
Enables the Tx Frequency test as part of an MI-BASIC test program

:TXFreq

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TXFreq:Status?`

Example response : `PA:TXF:S ON`

:PAram

Applicable to MPT1327 and PMR

:TXLEvel

:Averages

Description : Sets the number of averages for the test.
Parameter : <NR1> Range 0 to 99
Allowed suffices : Not applicable
Example : `:PAram:TXLEvel:Averages 10`
Sets the number of averages to 10

:TXLEvel

:Averages?

Description : Returns currently set value.
Response : <NR1> Range 0 to 99
Example : `:PAram:TXLEvel:Averages?`
Example response : `PA:TXLE:A 10`

:PAram

Applicable to MPT1327 and PMR

:TXLEvel

:Lower

Description : Sets the lower received RF level limit for the test.
Parameter : <NRf> Range –135 to 48.75
Allowed suffices : `DBM`
Example : `:PAram:TXLEvel:Lower 30dBm`
Sets the lower received RF level to +30 dBm

:TXLEvel

:Lower?

Description : Returns currently set value.
Response : <NR2> Range –135 to 48.75
Example : `:PAram:TXLEvel:Lower?`
Example response : `PA:TXLE:L 30.0DBM`

:PAram

Applicable to MPT1327 and PMR

:TXLEvel

:Status

Description : Enables or disables the Tx Level test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TXLEvel:Status ON`
Enables the Tx Level test as part of an MI-BASIC test program

:TXLEvel

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TXLEvel:Status?`

Example response : `PA:TXLE:S ON`

:PAram

Applicable to MPT1327 and PMR

:TXLEvel

:Upper

Description : Sets the upper received RF level limit for the test.

Parameter : <NRf> Range –135 to 48.75

Allowed suffices : `DBM`

Example : `:PAram:TXLEvel:Upper 40dBm`
Sets the upper received RF level to +40 dBm

:TXLEvel

:Upper?

Description : Returns currently set value.

Response : <NR2> Range –135 to 48.75

Example : `:PAram:TXLEvel:Upper?`

Example response : `PA:TXLE:U 40.0DBM`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXLImit

:Averages

Description : Sets the number of averages for the test.
Parameter : <NR1> Range 0 to 99
Allowed suffices : Not applicable
Example : `:PAram:TXLImit:Averages 10`
Sets the number of averages to 10

:TXLImit

:Averages?

Description : Returns currently set value.
Response : <NR1> Range 0 to 99
Example : `:PAram:TXLImit:Averages?`
Example response : `PA:TXLI:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXLImit

:Modlevel

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `PAram:TXLImit:Modlevel 5.7kHz`
Sets the test's reference modulation level to 5.7 kHz

:TXLImit

:Modlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:TXLImit:Modlevel?`
Example response : `PA:TXLI:M 5700.0HZ`

:PAram

Applicable to NAMPS and NTACS

:TXLImit

:NBModlevel

Description : Sets the reference narrow-band modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `PAram:TXLImit:NBModlevel 3kHz`
Sets the test's narrow-band reference modulation level to 3 kHz

:TXLImit

:NBModlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:TXLImit:NBModlevel?`
Example response : `PA:TXLI:NBM 3000.0HZ`

:PAram

Applicable to NAMPS and NTACS

:TXLImit

:NBUpper

Description : Sets the upper deviation limit for the test for narrow-band systems.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `PAram:TXLImit:NBUpper 5kHz`
Sets the narrow-band upper deviation limit for the test to 5 kHz

:TXLImit

:NBUpper?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:TXLImit:NBUpper?`
Example response : `PA:TXLI:NBU 5000.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXLImit

:Overload

Description : Sets the overload factor for the test.

Parameter : <NRf> Range 0 to 100

Allowed suffices : DB

Example : `:PAram:TXLImit:Overload 20dB`
Sets the Tx Limiting test overload factor to 20 dB

:TXLImit

:Overload?

Description : Returns currently set value.

Response : <NR2> Range 0 to 100

Example : `:PAram:TXLImit:Overload?`

Example response : `PA:TXLI:O 20.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXLImit

:Status

Description : Enables or disables the Tx Limiting test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TXLImit:Status ON`
Enables the Tx Limiting test as part of an MI-BASIC test program

:TXLImit

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TXLImit:Status?`

Example response : `PA:TXLI:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXLImit

:Txfilter

Description : Sets the demodulation filter type for the Rx Expansion test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:TXLImit:Txfilter CCITT`
Sets the demodulation filter to CCITT

:TXLImit

:Txfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:TXLImit:Txfilter?`

Example response : `PA:TXLI:T CCITT`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXLImit

:Upper

Description : Sets the upper deviation limit for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `:PAram:TXLImit:Upper 9.5kHz`
Sets the deviation limit for the test to 9.5 kHz

:TXLImit

:Upper?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : `:PAram:TXLImit:Upper?`
Example response : `PA:TXLI:U 9500.0HZ`

:PAram

Applicable to PMR

:TXModsens

:Lower

Description : Sets the lower AF voltage limit for the test.
Parameter : <NRf> Range 0 to 5 V RMS
Allowed suffices : `UV, MV, V`
Example : `:PAram:TXModsens:Lower 10MV`
Sets the deviation lower limit for the test to 10 mV

:TXModsens

:Lower?

Description : Returns currently set value.
Response : <NR2> Range 0 to 5 V RMS
Example : `:PAram:TXModsens:Lower?`
Example response : `PA:TXM:L 10.0MV`

:PAram

Applicable to PMR

:TXModsens

:Modlevel

Description : Sets the reference modulation level for the test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:TXModsens:Modlevel 1kHz`
Sets the test's reference modulation level to 1 kHz

:TXModsens

:Modlevel?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:TXModsens:Modlevel?`

Example response : `PA:TXM:M 1000.0HZ`

:PAram

Applicable to PMR

:TXModsens

:Status

Description : Enables or disables the Tx Mod Sensitivity test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TXModsens:Status ON`
Enables the Tx Mod Sensitivity test as part of an MI-BASIC test program

:TXModsens

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TXModsens:Status?`

Example response : `PA:TXM:S ON`

:PAram

Applicable to PMR

:TXModsens

:Txfilter

Description : Sets the demodulation filter type for the Tx Mod Sensitivity test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:TXModsens:Txfilter CCITT`
Sets the demodulation filter to CCITT

:TXModsens

:Txfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:TXModsens:Txfilter?`

Example response : `PA:TXM:T CCITT`

:PAram

Applicable to PMR

:TXModsens

:Upper

Description : Sets the upper AF voltage limit for the test.
Parameter : <NRf> Range 0 to 5 V RMS
Allowed suffices : `UV, MV, V`
Example : `:PAram:TXModsens:Upper 5V`
Sets the deviation upper limit for the test to 5 V

:TXModsens

:Upper?

Description : Returns currently set value.
Response : <NR2> Range 0 to 5 V RMS
Example : `:PAram:TXModsens:Upper?`
Example response : `PA:TXM:U 5000.0MV`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXNoise

:Averages

Description : Sets the number of averages for the test.
Parameter : <NR1> Range 0 to 99
Allowed suffices : Not applicable
Example : `:PAram:TXNoise:Averages 10`
Sets the number of averages to 10

:TXNoise

:Averages?

Description : Returns currently set value.
Response : <NR1> Range 0 to 99
Example : `:PAram:TXNoise:Averages?`
Example response : `PA:TXN:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXNoise

:Status

Description : Enables or disables the Tx Noise test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TXNoise:Status ON`
Enables the Tx Noise test as part of an MI-BASIC test program

:TXNoise

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TXNoise:Status?`

Example response : `PA:TXN:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXNoise

:Txfilter

Description : Sets the demodulation filter type for the Tx Noise test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:TXNoise:Txfilter LP300HZ`
Sets the demodulation filter to 300 Hz low-pass

:TXNoise

:Txfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:TXNoise:Txfilter?`

Example response : `PA:TXN:T LP300HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXNoise

:Upper

Description : Sets the upper noise (residual FM) limit for the test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `PAram:TXNoise:Upper 300Hz`
Sets the upper noise limit for the test to 300 Hz

:TXNoise

:Upper?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:TXNoise:Upper?`

Example response : `PA:TXN:U 300.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSInad

:Averages

Description : Sets the number of averages for the test.

Parameter : <NR1> Range 0 to 99

Allowed suffices : Not applicable

Example : `:PAram:TXSInad:Averages 10`
Sets the number of averages to 10

:TXSInad

:Averages?

Description : Returns currently set value.

Response : <NR1> Range 0 to 99

Example : `:PAram:TXSInad:Averages?`

Example response : `PA:TXSI:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSInad

:Lower

Description : Sets the lower pass limit for the test.
Parameter : <NRf> Range 0 to 100
Allowed suffices : DB
Example : :PAram:TXSInad:Lower 26dB
Sets the lower pass limit to 26 dB

:TXSInad

:Lower?

Description : Returns currently set value.
Response : <NR2> Range 0 to 100
Example : :PAram:TXSInad:Lower?
Example response : PA:TXSI:L 26.0DB

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSInad

:Modlevel

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : :PAram:TXSInad:Modlevel 2.3kHz
Sets the test's reference modulation level to 2.3 kHz

:TXSInad

:Modlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : :PAram:TXSInad:Modlevel?
Example response : PA:TXSI:M 2300.0HZ

:PAram

Applicable to NAMPS and NTACS.

:TXSInad

:Nbmodlevel

Description : Sets the reference narrow-band modulation level for the test.

Parameter : <NRf> Range 0 to 40 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:PAram:TXSInad:Nbmodlevel 1.5kHz`
Sets the test's narrow-band reference modulation level to 1.5 kHz

:TXSInad

:Nbmodlevel?

Description : Returns currently set value.

Response : <NR2> Range 0 to 40 kHz

Example : `:PAram:TXSInad:Nbmodlevel?`

Example response : `PA:TXSI:N 1500.0HZ`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSInad

:Status

Description : Enables or disables the Tx SINAD test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TXSInad:Status ON`
Enables the Tx SINAD test as part of an MI-BASIC test program

:TXSInad

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TXSInad:Status?`

Example response : `PA:TXSI:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSInad

:Txfilter

Description : Sets the demodulation filter type for the Tx SINAD test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:TXSInad:Txfilter CCITT`
Sets the demodulation filter to CCITT

:TXSInad

:Txfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:TXSInad:Txfilter?`

Example response : `PA:TXSI:T CCITT`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSN

:Averages

Description : Sets the number of averages for the test.

Parameter : <NR1> Range 0 to 99

Allowed suffices : Not applicable

Example : `:PAram:TXSN:Averages 10`
Sets the number of averages to 10

:TXSN

:Averages?

Description : Returns currently set value.

Response : <NR1> Range 0 to 99

Example : `:PAram:TXSN:Averages?`

Example response : `PA:TXSN:A 10`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSN

:Lower

Description : Sets the lower pass limit for the test.

Parameter : <NRf> Range 0 to 1000

Allowed suffices : `DB`

Example : `:PAram:TXSN:Lower 26dB`
Sets the lower pass limit to 26 dB

:TXSN

:Lower?

Description : Returns currently set value.

Response : <NR2> Range 0 to 1000

Example : `:PAram:TXSN:Lower?`

Example response : `PA:TXSN:L 26.0DB`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSN

:Modlevel

Description : Sets the reference modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : :PAram:TXSN:Modlevel 2.3kHz
Sets the test's reference modulation level to 2.3 kHz

:TXSN

:Modlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : :PAram:TXSN:Modlevel?
Example response : PA:TXSN:M 2300.0HZ

:PAram

Applicable to NAMPS and NTACS

:TXSN

:Nbmodlevel

Description : Sets the reference narrow-band modulation level for the test.
Parameter : <NRf> Range 0 to 40 kHz
Allowed suffices : HZ, KHZ, MHZ
Example : :PAram:TXSN:Nbmodlevel 1.5kHz
Sets the test's reference modulation level to 1.5 kHz

:TXSN

:Nbmodlevel?

Description : Returns currently set value.
Response : <NR2> Range 0 to 40 kHz
Example : :PAram:TXSN:Nbmodlevel?
Example response : PA:TXSN:N 1500.0HZ

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSN

:Status

Description : Enables or disables the Tx S/N test as part of an MI-BASIC test program.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:PAram:TXSN:Status ON`
Enables the Tx S/N test as part of an MI-BASIC test program

:TXSN

:Status?

Description : Returns currently set value.

Response : <CRD> OFF, ON

Example : `:PAram:TXSN:Status?`

Example response : `PA:TXSN:S ON`

:PAram

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327 and PMR

:TXSN

:Txfilter

Description : Sets the demodulation filter type for the Tx S/N test.

Parameter : <CPD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Allowed suffices : Not applicable

Example : `:PAram:TXSN:Txfilter CCITT`
Sets the demodulation filter to CCITT

:TXSN

:Txfilter?

Description : Returns currently set value.

Response : <CRD> NONE
LP300HZ
LP5KHZ
LP20KHZ
STDBP
CCITT
CMESS

Example : `:PAram:TXSN:Txfilter?`

Example response : `PA:TXSN:T CCITT`

Control commands (:SYCtrl)

List of commands

:SYCtrl
:Speechq

:SYCtrl

Applicable to GSM, DCS and PCS

:Speechq

Description : Controls the result of the speech quality test (:STEst:SPeechqual, page 6-21). Emulates pressing *[pass test]* or *[fail test]* soft key.

Parameter : <CPD> PASS or FAIL

Allowed suffices : Not applicable

Example : `:SYCtrl:Speechq PASS`
Pass the speech quality test

Manual commands (:SYManual)

Introduction

This section describes Systems Manual commands, that is, those commands that are applicable to Systems Manual test settings.

If Systems verbose mode is ON (see SYSetup:VErbosemode command) the Systems Manual screen will be updated.

To obtain Measurement results in Systems Manual test mode use the MEASDig: query commands described in chapter 5.

List of commands

:SYManual
- **:Burst**
  - **:T1mode**
  - **:T1mode?**
  - **:TYpe**
  - **:TYpe?**

:Channel
- **:Control**
- **:Control?**
- **:Traffic**
- **:Traffic?**
- **:Slot**
- **:Slot?**

:Mode
:Mode?
:Rflevel
:Rflevel?
:T1type
:T1type?
:TChtype
:TChtype?

:SYManual

Applicable to TETRA MOBILE

:Burst

:T1mode

Description : Signals a mobile what type of bursts to transmit in T1 test mode

Parameter : <CPD> NORMAL, CONTROL or NONE

Allowed suffices : Not applicable

Example : `:SYManual:Burst:T1mode NONE`
Signal the mobile not to transmit any bursts

:Burst

:T1mode?

Description : Returns currently set value.

Response : <CRD> NORMAL, CONTROL or NONE

Example : `:SYManual:Burst:T1mode?`

Example response : `SYM:B:T1 CONTROL`

:SYManual

Applicable to TETRA MOBILE, TETRA BASE and TETRA DIRECT

:Burst

:TYpe

Description : Sets the type of bursts to be measured

Parameter : <CPD> NORMAL, CONTROL or NORMAL_CONTROL
(for TETRA MOBILE in MCCH or TCH Test mode)

or NORMAL, SYNC or NORMAL_SYNC
(for TETRA BASE and TETRA DIRECT)

Allowed suffices : Not applicable

Example : `:SYManual:Burst:TYpe NORMAL`
Set the type of bursts to be measured to normal bursts

:Burst

:TYpe?

Description : Returns currently set value.

Response : <CRD> NORMAL, CONTROL or NORMAL_CONTROL
(for TETRA MOBILE)

or NORMAL, SYNC or NORMAL_SYNC
(for TETRA BASE and TETRA DIRECT)

Example : `:SYManual:Burst:TYpe?`

Example response : `SYM:B:TY CONTROL`

:SYManual

Applicable to TETRA MOBILE, TETRA BASE and TETRA DIRECT

:Channel

:Control

Description : Sets the control channel (or the channel for TETRA DIRECT)
Parameter : <NR1> Range 0 to 9999 depending on selected system
Allowed suffices : Not applicable
Example : `:SYManual:Channel:Control 50`
Set the control channel to channel 50

:Channel

:Control?

Description : Returns currently set value.
Response : <NR1> Range 0 to 9999 depending on selected system
Example : `:SYManual:Channel:Control?`
Example response : `SYM:C:C 50`

:SYManual

Applicable to TETRA MOBILE

:Channel

:Traffic

Description : Sets the traffic channel
Parameter : <NR1> Range 0 to 9999 depending on selected system
Allowed suffices : Not applicable
Example : `:SYManual:Channel:Traffic 50`
Set the Traffic channel to channel 50

:Channel

:Traffic?

Description : Returns currently set value.
Response : <NR1> Range 0 to 9999 depending on selected system
Example : `:SYManual:Channel:Traffic?`
Example response : `SYM:C:T 50`

:SYManual

Applicable to TETRA MOBILE

:Channel

:Slot

Description : Sets the traffic channel timeslot
Parameter : <NR1>Range 1 to 4
Allowed suffices : Not applicable
Example : `:SYManual:Channel:Slot 1`
Set the traffic channel timeslot to 1

:Channel

:Slot?

Description : Returns currently set value.
Response : <NR1>Range 1 to 4
Example : `:SYManual:Channel:Slot?`
Example response : `SYM:C:S 1`

:SYManual

Applicable to TETRA MOBILE

:Mode

Description : Sets the Manual test mode
Parameter : <CPD> MCCH, TCHTEST or T1TEST
Allowed suffices : Not applicable
Example : `:SYManual:Mode MCCH`
Sets the manual test mode to MCCH

:Mode?

Description : Returns currently set value.
Response : <CRD> MCCH, TCHTEST or T1TEST
Example : `:SYManual:Mode?`
Example response : `SYM:M MCCH`

:SYManual

Applicable to TETRA MOBILE

:Rflevel

Description : Sets the output level of the test set RF generator.
Parameter : <NRf>
Allowed suffices : DBM
Example : `:SYManual:Rflevel -80DBM`
Set the RF generator output level to -80dBm

:Rflevel?

Description : Returns currently set value.
Response : <NR2> Units in dBm
Example : `:SYManual:Rflevel?`
Example response : `SYM:R -80DBM`

:SYManual

Applicable to TETRA MOBILE

:T1type

Description : Sets the channel type for the T1 test mode
Parameter : <CPD> TCH_72, TCH_24, or SCH_F
Allowed suffices : Not applicable
Example : `:SYManual:T1test TCH_72`
Set channel type for the T1 test mode to TCH/7.2

:T1type?

Description : Returns currently set value.
Response : <CRD> TCH_72, TCH_24, or SCH_F
Example : `:SYManual:T1type?`
Example response : `SYM:T1 TCH_72`

:SYManual

Applicable to TETRA MOBILE

:TChtype

Description : Sets the channel type for the TCH test mode

Parameter : <CPD> TCH_72 or TCH_24

Allowed suffices : Not applicable

Example : `:SYManual:TChtest TCH_72`

Set channel type for the T1 test mode to TCH/7.2

:TChtype?

Description : Returns currently set value.

Response : <CRD> TCH_72 or TCH_24

Example : `:SYManual:TChtype?`

`SYM:TC TCH_72`

Setup commands (:SYSetup)

List of commands

:SYSetup
- :ACcessparam
- :ACcessparam?
- :AId
- :AId?
- :ALohan
- :ALohan?
- :AReanumber
- :AReanumber?
- :BASEService
- :BASEService?
- :BASETxmode
- :BASETxmode?
- :BCAstn
  - :Include n,
  - :Include? n
  - :Parameters n,
  - :Parameters? n
  - :Sysdef n,
  - :Sysdef? n
- :BCC
- :BCC?
- :CC
- :CC?
- :CDdc
- :CDdc?
- :CEllbar
- :CEllbar?
- :CGChaccess
- :CGChaccess?
- :CGDc
- :CGDc?
- :CHargerate
- :CHargerate?
- :CIdent
- :CIdent?
- :COmpandor
- :COmpandor?
- :CPrefix
- :CPrefix?
- :DCc
- :DCc?
- :DIsplay
- :DIsplay?
- :DScc
- :DScc?
- :FOa
- :FOa?
- :FTC
- :FTC?
- :FTCS
- :FTCS?
- :FVc
- :FVc?
- :GIdent
- :GIdent?
- :GPrefix
- :GPrefix?
- :HANDOFFINc
- :HANDOFFINc?
- :HANDOFFIS
- :HANDOFFIS?
- :HANDOFFScheme
- :HANDOFFScheme?
- :HCcb
- :HCcb?
- :HTcb
- :HTcb?
- :IDstep
- :IDstep?
- :IMSI
- :IMSI?
- :IMSIAttach
- :IMSIAttach?
- :ITsiattach
- :ITsiattach?
- :LAc
- :LAc?
- :LCcb
- :LCcb?
- :LTC
- :LTC?
- :LTCB
- :LTCB?
- :LTCS
- :LTCS?
- :LVc
- :LVc?
- :MAxtxlev
- :MAxtxlev?
- :MCc
- :MCc?
- :MIDent
- :MIDent?
- :MIN
- :MIN?
- :MINRxlev
- :MINRxlev?
- :MNc
- :MNc?
- :MObiletype
- :MObiletype?
- :MPrefix
- :MPrefix?
- :NCc
- :NCc?
- :NFormat
- :NFormat?
- :PAusemode
- :PAusemode?
- :PRint
- :PRint?
- :RAND1
- :RAND1?
- :RAND2
- :RAND2?
- :RAND3
- :RAND3?
- :RAND4
- :RAND4?
- :REGID
- :REGID?
- :REGINc
- :REGINc?
- :RESp
- :RESp?
- :SAtphi
- :SAtphi?
- :SCAnband
- :SCAnband?
- :SCC
- :SCC?
- :SID
- :SID?
- :SIGnallingscheme
- :SIGnallingscheme?
- :SISc
- :SISc?
- :SPWD1
  - :Ep
  - :Ep?
  - :Pins
  - :Pins?
  - :REllevel
  - :REllevel?
  - :RSVD2

:RSVD2?
:RSVD3
:RSVD3?
:SPWD2
:CMax
:CMax?
:CPa
:CPa?
:Dtx
:Dtx?
:E
:E?
:Freg
:Freg?
:Npaging
:Npaging?
:RCf
:RCf?
:REGH
:REGH?
:REGR
:REGR?
:RSvd
:RSvd?
:S
:S?
:TA
:TA?
:TC
:TC?
:TCHtsc
:TCHtsc?
:TCOn
:TCOn?
:TYpe
:TYpe?
:VAriant
:VAriant?
:VErbosemode
:VErbosemode?

:SYSetup

Applicable to TETRA MOBILE

:ACcessparam

Description : Sets the mobile's access parameter.

Parameter : <NR2> Range –23dBm to –53dBm, rounded up to the nearest 2dB step.

Allowed suffices : DBM

Example : `:SYSetup:ACcessparam –24DBM`
Sets the mobile's access parameterl to –23 dBm (rounded up)

:ACcessparam?

Description : Returns currently set value.

Response : <NR2> Range –23 dBm to –53 dBm in 2dB steps.

Example : `:SYSetup:ACcessparam?`

Example response : `SYS:AC –23DBM`

:SYSetup

Applicable to TACS

:AId

Description : Sets TACS Area Identification.
Parameter : <NR1> Range 0 to 32767
Allowed suffices : Not applicable
Example : `:SYSetup:AId 12345`
Sets area identification number to 12345

:AId?

Description : Returns currently set value.
Response : <NR1> Range 0 to 32767
Example : `:SYSetup:AId?`
Example response : `SYS:AI 12345`

:SYSetup

Applicable to MPT1327

:ALohan

Description : Sets size of ALOHA frame for MPT1327.
Parameter : <NR1> Range 1 to 15
Allowed suffices : Not applicable
Example : `:SYSetup:ALohan 5`
Sets the ALOHA frame length to 5 slots

:ALohan?

Description : Returns currently set value.
Response : <NR1> Range 1 to 15
Example : `:SYSetup:ALohan?`
Example response : `SYS:AL 5`

:SYSetup

Applicable to NMT

:AReanumber

Description : Sets NMT Area Number.
Parameter : <NR1> Range 0 to 4
Allowed suffices : Not applicable
Example : `:SYSetup:AReanumber 3`
Sets area number to 3

:AReanumber?

Description : Returns currently set value.
Response : <NR1> Range 0 to 4
Example : `:SYSetup:AReanumber?`
Example response : `SYS:AR 3`

:SYSetup

Applicable to TETRA MOBILE

:BASEService

Description : Sets the mobile's Base Station Service field
Parameter : <CPD> NOT_SUPPORTED or SUPPORTED
Allowed suffices : Not applicable
Example : `:SYSetup:BASEService NOT_SUPPORTED`
Sets the mobile's Base Station Service field to none supported

:BASEService?

Description : Returns currently set value.
Response : <CRD> NOT_SUPPORTED or SUPPORTED
Example : `:SYSetup:BASEService?`
Example response : `SYS:BASES NOT_SUPPORTED`

:SYSetup

Applicable to TETRA BASE

:BASETxmode

Description : Sets the test set to match the Base Station's transmit mode

Parameter : <CPD> DISCONTINUOUS or
CONTINUOUS_ALL_SLOTS

Allowed suffices : Not applicable

Example : `:SYSetup:BASETxmode CONTINUOUS_ALL_SLOTS`
Sets the test set to measure the signal from a Base Station transmitting in continuoas mode on all timeslots

:BASETxmode?

Description : Returns currently set value.

Response : <CRD> DISCONTINUOUS or
CONTINUOUS_ALL_SLOTS

Example : `:SYSetup:BASETxmode?`

Example response : `SYS:BASET CONTINUOUS_ALL_SLOTS`

:SYSetup

Applicable to MPT1327

:BCAstn

:Include *n*,

Description : Includes or excludes Broadcast Message specified by the number *n*.

Parameters : <NR1> *n* = Broadcast Message number 1 to 4
<CPD> NO or YES

Allowed suffices : Not applicable

Example : `:SYSetup:BCAstn:Include 3,YES`
Now Broadcast Message type 3 is included in ALOHA frame

:BCAstn

:Include? *n*

Description : Returns Include/Exclude state for Broadcast Message specified by the number *n*.

Response : <CRD> NO or YES

Parameter : <NR1> *n* = Broadcast Message number 1 to 4

Example : `:SYSetup:BCAstn:Include? 3`

Example response : `SYS:BCA:I YES`

:SYSetup

Applicable to MPT1327

:BCAstn

:Parameters *n*,

Description : Sets the Parameters field of the Broadcast Message specified by the number *n*.

Parameters : <NR1> *n* = Broadcast Message number 1 to 4
<NR1> Parameters field, range 0 to #H3FFFF

Allowed suffices : Not applicable

Example : `SYSetup:BCAstn:Parameters 2, #H12`
Sets Broadcast Message number 2's Parameters field to #H12

:BCAstn

:Parameters? *n*

Description : Returns the contents of the Parameters field for the Broadcast Message specified by the number *n*.

Response : <NR1> Range 0 to #H3FFFF

Parameter : <NR1> *n* = Broadcast Message number 1 to 4

Example : `:SYSetup:BCAstn:Parameters? 2`

Example response : `SYS:BCA:P #H12`

:SYSetup

Applicable to MPT1327

:BCAstn

:Sysdef *n*,

Description : Sets the System Definition field of the Broadcast Message specified by the number *n*.

Parameters : <NR1> *n* = Broadcast Message number 1 to 4
<NR1> System Definition number, range 0 to #H1F

Allowed suffices : Not applicable

Example : `:SYSetup:BCAstn:Sysdef 2,#H12`
Sets Broadcast Message number 2's Sysdef field to #H12

:BCAstn

:Sysdef? *n*

Description : Returns the contents of the System Definition field for the Broadcast Message specified by the number *n*.

Response : <NR1> Range 0 to #H1F

Parameter : <NR1> *n* = Broadcast Message number 1 to 4

Example : `:SYSetup:BCAstn:Sysdef? 2`

Example response : `SYS:BCA:S #H12`

:SYSetup

Applicable to GSM, DCS, PCS and TETRA MOBILE

:BCC

Description : Sets the Base Colour Code.

For GSM, DCS and PCS, also set the BCCH training sequence.

Parameter : <NR1> Range 0 to 7 for GSM, DCS and PCS
or 0 to 63 for TETRA MOBILE

Allowed suffices : Not applicable

Example : `:SYSetup:BCC 4`
Sets the Base Colour Code to 4

:BCC?

Description : Returns currently set value.

Response : <NR1> Range 0 to 7 for GSM, DCS and PCS
or 0 to 63 for TETRA MOBILE

Example : `:SYSetup:BCC?`

Example response : `SYS:BCC 4`

:SYSetup

Applicable to AMPS, NAMPS, TACS, NTACS, NMT, MPT1327, GSM, DCS, PCS, TETRA MOBILE, TETRA BASE and TETRA DIRECT

:CC

Description : For AMPS, NAMPS, TACS, NTACS, NMT, MPT1327, GSM, DCS and PCS, sets the default Control Channel for the selected system to be used during a test program.
For TETRA MOBILE and TETRA BASE, sets the default Control Channel used in Manual Test.
For TETRA DIRECT, sets the default Channel used in Manual Test.

Parameter : <NR1> Range 0 to 9999 depending on selected system

Allowed suffices : Not applicable

Example : `:SYSetup:CC 100`
Sets the Control Channel number to 100

:CC?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999 depending on the selected system

Example : `:SYSetup:CC?`

Example response : `SYS:CC 100`

:SYSetup

Applicable to MPT1327

:CDdc

Description : Sets an MPT1327 mobile's Called Disconnect count.

Parameter : <NR1> Range 0 to 9

Allowed suffices : Not applicable

Example : `:SYSetup:CDdc 5`
Sets the Called Disconnect parameter to 5

:CDdc?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9

Example : `:SYSetup:CDdc?`

Example response : `SYS:CD 5`

:SYSetup

Applicable to GSM, DCS and PCS

:CEllbar

Description : Sets the mobile's Cell Barred bit

Parameter : <CPD> NOT_BARRED or BARRED

Allowed suffices : Not applicable

Example : `:SYSetup:CEllbar NOT_BARRED`
Clears the mobile's Cell Barred bit

:CEllbar?

Description : Returns currently set value.

Response : <CRD> NOT_BARRED or BARRED

Example : `:SYSetup:CEllbar?`

Example response : `SYS:CE NOT_BARRED`

:SYSetup

Applicable to NMT

:CGChaccess

Description : Sets an NMT900 system's calling access channel.

Parameter : <CPD > ALL, GROUP_A or GROUP_B

Allowed suffices : Not applicable

Example : `:SYSetup:CGChaccess GROUP_B`
Sets the calling access channel to GROUP_B

:CGChaccess?

Description : Returns currently set value.

Response : <CRD> ALL, GROUP_A or GROUP_B

Example : `:SYSetup:CGChaccess?`

Example response : `SYS:CGC GROUP_B`

:SYSetup

Applicable to MPT1327

:CGDc

Description : Sets an MPT1327 mobile's Calling Disconnect count.
Parameter : <NR1> Range 0 to 9
Allowed suffices : Not applicable
Example : `:SYSetup:CGDc 5`
Sets the Calling Disconnect count to 5

:CGDc?

Description : Returns currently set value.
Response : <NR1> Range 0 to 9
Example : `:SYSetup:CGDc?`
Example response : `SYS:CGD 5`

:SYSetup

Applicable to TACS

:CHargerate

Description : Sets a TACS mobile's charge rate.
Parameter : <NR1> Range 0 to 2047
Allowed suffices : Not applicable
Example : `:SYSetup:CHargerate 5`
Sets the charge rate to 5

:CHargerate?

Description : Returns currently set value.
Response : <NR1> Range 0 to 2047
Example : `:SYSetup:CHargerate?`
Example response : `SYS:CH 5`

:SYSetup

Applicable to MPT1327

:CIdent

Description : Sets an MPT1327 mobile's Calling Identity.
Parameter : <NR1> Range 0 to 8191
Allowed suffices : Not applicable
Example : `:SYSetup:CIdent 2000`
Sets the calling identity to 2000

:CIdent?

Description : Returns currently set value.
Response : <NR1> Range 0 to 8191
Example : `:SYSetup:CIdent?`
Example response : `SYS:CI 2000`

:SYSetup

Applicable to NMT

:COmpandor

Description : Requests the NMT450i mobile's compandor to be in or out.
Parameter : <CPD > OUT or IN
Allowed suffices : Not applicable
Example : `:SYSetup:COmpandor IN`
Sets the NMT450i mobile's compandor to be IN

:COmpandor?

Description : Returns currently set value.
Response : <CPD> OUT or IN
Example : `:SYSetup:COmpandor?`
Example response : `SYS:CO IN`

:SYSetup

Applicable to MPT1327

:CPrefix

Description : Sets an MPT1327 mobile's Calling Prefix.
Parameter : <NR1> Range 0 to 127
Allowed suffices : Not applicable
Example : `:SYSetup:CPrefix 100`
Sets the Calling Prefix to 100

:CPrefix?

Description : Returns currently set value.
Response : <NR1> Range 0 to 127
Example : `:SYSetup:CPrefix?`
Example response : `SYS:CP 100`

:SYSetup

Applicable to AMPS and TACS

:DCc

Description : Sets the system's Digital Colour Code.
Parameter : <NR1> Range 0 to 3
Allowed suffices : Not applicable
Example : `:SYSetup:DCc 2`
Sets the Digital Colour Code to 2

:DCc?

Description : Returns currently set value.
Response : <NR1> Range 0 to 3
Example : `:SYSetup:DCc?`
Example response : `SYS:DC 2`

:SYSetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:DIsplay

Description : Sets the Autorun control display format to Summary or Full.

Parameter : <CPD > SUMMARY or FULL

Allowed suffices : Not applicable

Example : `:SYSetup:DIsplay SUMMARY`
Autorun tests will now be displayed in Summary format

:DIsplay?

Description : Returns currently set value.

Response : <CRD> SUMMARY or FULL

Example : `:SYSetup:DIsplay?`

Example response : `SYS:DI SUMMARY`

:SYSetup

Applicable to NAMPS and NTACS

:DScc

Description : Sets the system's Digital SAT Colour Code.

Parameter : <NR1> Range 0 to 6

Allowed suffices : Not applicable

Example : `:SYSetup:DScc 2`
Sets the Digital SAT Colour Code to 2

:DScc?

Description : Returns currently set value.

Response : <NR1> Range 0 to 6

Example : `:SYSetup:DScc?`

Example response : `SYS:DS 2`

:SYSetup

Applicable to MPT1327

:FOa

Description : Enables or disables full off-air signalling for MPT1327.

Parameter : <CPD> DISABLE or ENABLE

Allowed suffices : Not applicable

Example : `:SYSetup:FOa ENABLE`
Enables full off-air signalling for MPT1327 system

:FOa?

Description : Returns currently set value.

Response : <CRD> DISABLE or ENABLE

Example : `:SYSetup:FOa?`

Example response : `SYS:FO ENABLE`

:SYSetup

Applicable to MPT1327, NMT, PMR, GSM, DCS, PCS and TETRA MOBILE

:FTC

Description : For, MPT1327, NMT, PMR, GSM, DCS and PCS, sets the First Traffic Channel to be used by a test program.
For TETRA MOBILE, sets the default Traffic Channel used in Manual Test.

Parameter : <NR1> Range 0 to 9999 depending on system

Allowed suffices : Not applicable

Example : `:SYSetup:FTC 100`
Sets the First Traffic Channel 100

:FTC?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999 depending on selected system

Example : `:SYSetup:FTC?`

Example response : `SYS:FTC 100`

:SYSetup

Applicable to GSM, DCS, PCS and TETRA MOBILE

:FTCS

Description : For GSM, DCS and PCS, sets the First Traffic Channel Timeslot to be used by a test program.
For TETRA MOBILE, sets the default Traffic Channel Timeslot used in Manual Test.

Parameter : <NR1> Range 0 to 7 (for GSM, DCS & PCS)
Range 1 to 4 (for TETRA MOBILE)

Allowed suffices : Not applicable

Example : `:SYSetup:FTCS 2`
Sets the First Traffic Channel Timeslot to 2

:FTCS?

Description : Returns currently set value.

Response : <NR1> Range 0 to 7 (for GSM, DCS & PCS)
Range 1 to 4 (for TETRA MOBILE)

Example : `:SYSetup:FTCS?`

Example response : `SYS:FTCS 2`

:SYSetup

Applicable to AMPS, NAMPS, PMR, TACS and NTACS

:FVc

Description : Sets the First Voice Channel to be used by a test program.

Parameter : <NR1> Range 0 to 9999 depending on system

Allowed suffices : Not applicable

Example : `:SYSetup:FVc 100`
Sets the First Voice Channel to 100

:FVc?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999 depending on selected system

Example : `:SYSetup:FVc?`

Example response : `SYS:FV 100`

:SYSetup

Applicable to MPT1327

:GIdent

Description : Sets an MPT1327 mobile's Group Calling Identity.
Parameter : <NR1> Range 0 to 8191
Allowed suffices : Not applicable
Example : `:SYSetup:GIdent 100`
Sets the Group Calling Identity to 100

:GIdent?

Description : Returns currently set value.
Response : <NR1> Range 0 to 8191
Example : `:SYSetup:GIdent?`
Example response : `SYS:GI 100`

:SYSetup

Applicable to MPT1327

:GPrefix

Description : Sets an MPT1327 mobile's Group Calling Prefix.
Parameter : <NR1> Range 0 to 127
Allowed suffices : Not applicable
Example : `:SYSetup:GPrefix 100`
Sets the Group Calling Prefix to 100

:GPrefix?

Description : Returns currently set value.
Response : <NR1> Range 0 to 127
Example : `:SYSetup:GPrefix?`
Example response : `SYS:GP 100`

:SYSetup

Applicable to AMPS, TACS, NAMPS, NTACS, NMT, PMR, MPT1327, GSM, DCS and PCS

:HANDOFFINc

Description : Sets the handoff increment to be used for the Handoff test in a test program.

Parameter : <NR1> Range 0 to 9999 depending on selected system

Allowed suffices : Not applicable

Example : `:SYSetup:HANDOFFINc 100`
Sets the handoff increment to 100

:HANDOFFINc?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999 depending on selected system

Example : `:SYSetup:HANDOFFINc?`

Example response : `SYS:HANDOFFIN 100`

:SYSetup

Applicable to GSM, DCS and PCS

:HANDOFFIS

Description : Sets the Handoff Incremental timeslot to be used for the Digital Handoff test in a test program.

Parameter : <NR1> Range 0 to 7

Allowed suffices : Not applicable

Example : `:SYSetup:HANDOFFIS 4`
Sets the Handoff Incremental timeslot to 4

:HANDOFFIS?

Description : Returns currently set value.

Response : <NR1> Range 0 to 7

Example : `:SYSetup:HANDOFFIS?`

Example response : `SYS:HANDOFFIS 4`

:SYSetup

Applicable to NMT

:HANDOFFScheme

Description : Sets the handoff scheme to be used.

Parameter : <CPD> C = 'ordinary' (NMT450, NMT450i, NMT900)
C1 = 'improved' (NMT900)
C2 = 'short' (NMT450i, NMT900)

Allowed suffices : Not applicable

Example : `:SYSetup:HANDOFFScheme C1`
Sets the handoff scheme to 'improved'

:HANDOFFScheme?

Description : Returns currently set value.

Response : <CRD> C, C1 or C2

Example : `:SYSetup:HANDOFFScheme?`

Example response : `SYS:HANDOFFS C1`

:SYSetup

Applicable to NMT

:HCcb

Description : Sets the higher control channel band.

Parameter : <NR1> Range 0 to 9999

Allowed suffices : Not applicable

Example : `:SYSetup:HCcb 1000`
Sets the higher control channel band to 1000

:HCcb?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999

Example : `:SYSetup:HCcb?`

Example response : `SYS:HC 1000`

:SYSetup

Applicable to NMT

:HTcb

Description : Sets the higher traffic channel band.
Parameter : <NR1> Range 0 to 9999
Allowed suffices : Not applicable
Example : `:SYSetup:HTcb 1000`
Sets the higher traffic channel band to 1000

:HTcb?

Description : Returns currently set value.
Response : <NR1> Range 0 to 9999
Example : `:SYSetup:HTcb?`
Example response : `SYS:HT 1000`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:IDstep

Description : Sets the identity step for these systems.
Parameter : <NR1> Range 0 to 65535
Allowed suffices : Not applicable
Example : `:SYSetup:IDstep 500`
Sets the identity step to 500

:IDstep?

Description : Returns currently set value.
Response : <NR1> Range 0 to 65536
Example : `:SYSetup:IDstep?`
Example response : `SYS:ID 500`

:SYSetup

Applicable to GSM, DCS and PCS

:IMSI

Description : Sets the International Mobile Subscriber Identity.

Parameter : <CPD> in the format "aaa-bb-ccccccc"
where aaa = decimal characters
bbb = decimal characters
cccc = 1 to 10 decimal characters

Allowed suffices : Not applicable

Examples :
```
:SYSetup:IMSI "001-01-0123456789"
:SYSetup:IMSI "001-01-3"
```
Sets the IMSI to 001-01-0123456789 or 001-01-3

:IMSI?

Description : Returns currently set value.

Response : <CRD> in the format "aaa-bb-cccccc"
where aaa = decimal characters
bb = decimal characters
cccc = 1 to 10 decimal characters

Example :
```
:SYSetup:IMSI?
```

Example responses :
```
SYS:IMSI "001-01-0123456789"
SYS:IMSI "001-01-3"
```

:SYSetup

Applicable to GSM, DCS and PCS

:IMSIAttach

Description : Sets the mobile's IMSI attach allowed flag.

Parameter : <CPD> NOT_ALLOWED or ALLOWED

Allowed suffices : Not applicable

Example :
```
:SYSetup:IMSIAttach ALLOWED
```
Sets the mobile's IMSI attach allowed flag

:IMSIAttach?

Description : Returns currently set value.

Response : <CRD> NOT_ALLOWED or ALLOWED

Example :
```
:SYSetup:IMSIAttach?
```

Example response :
```
SYS:IMSIA ALLOWED
```

:SYSetup

Applicable to TETRA MOBILE

:ITsiattach

Description : Sets the mobile's ITSI attach/detach required flag.

Parameter : <CPD> NOT_REQUIRED or REQUIRED

Allowed suffices : Not applicable

Example : `:SYSetup:ITsiattach REQUIRED`
Sets the mobile's ITSI attach required flag

:ITsiattach?

Description : Returns currently set value.

Response : <CRD> NOT_REQUIRED or REQUIRED

Example : `:SYSetup:ITsiattach?`

Example response : `SYS:IT REQUIRED`

:SYSetup

Applicable to GSM, DCS, PCS and TETRA MOBILE

:LAc

Description : Sets the Location Area Code.

Parameter : <NR1> Range 0 to 65535 for GSM, DCS and PCS
or 0 to 16383 for TETRA MOBILE

Allowed suffices : Not applicable

Example : `:SYSetup:LAc 400`
Sets the Location Area Code to 400

:LAc?

Description : Returns currently set value.

Response : <NR1> Range 0 to 65535 for GSM, DCS and PCS
or 0 to 16383 for TETRA MOBILE

Example : `:SYSetup:LAc?`

Example response : `SYS:LA 400`

:SYSetup

Applicable to NMT

:LCcb

Description : Sets the lower control channel band.

Parameter : <NR1> Range 0 to 9999

Allowed suffices : Not applicable

Example : `:SYSetup:LCcb 400`
Sets the lower scan band to 400

:LCcb?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999

Example : `:SYSetup:LCcb?`

Example response : `SYS:LC 400`

:SYSetup

Applicable to NMT, PMR, MPT1327, GSM, DCS and PCS

:LTC

Description : Sets the last traffic channel to be used for the Handoff test in a test program.

Parameter : <NR1> Range 0 to 9999 depending on selected system

Allowed suffices : Not applicable

Example : `:SYSetup:LTC 300`
Sets the last traffic channel to 300

:LTC?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999

Example : `:SYSetup:LTC?`

Example response : `SYS:LTC 300`

:SYSetup

Applicable to NMT

:LTCB

Description : Sets the lower traffic channel band.

Parameter : <NR1> Range 0 to 9999

Allowed suffices : Not applicable

Example : `:SYSetup:LTCB 400`
Sets the lower traffic channel band to 400

:LTCB?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999

Example : `:SYSetup:LTCB?`

Example response : `SYS:LTCB 400`

:SYSetup

Applicable to GSM, DCS and PCS

:LTCS

Description : Sets the Last Traffic Channel timeslot to be used for the Digital Handoff test in a test program.

Parameter : <NR1> Range 0 to 7

Allowed suffices : Not applicable

Example : `:SYSetup:LTCS 4`
Sets the Last Traffic Channel timeslot to 4

:LTCS?

Description : Returns currently set value.

Response : <NR1> Range 0 to 7

Example : `:SYSetup:LTCS?`

Example response : `SYS:LTCS: 4`

:SYSetup

Applicable to AMPS, NAMPS, PMR, TACS and NTACS

:LVc

Description : Sets the last voice channel to be used for the Handoff test in a test program.

Parameter : <NR1> Range 0 to 9999 depending on selected system

Allowed suffices : Not applicable

Example : `:SYSetup:LVc 300`
Sets the last voice channel to 300

:LVc?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999

Example : `:SYSetup:LVc?`

Example response : `SYS:LV 300`

:SYSetup

Applicable to TETRA MOBILE

:MAxtxlev

Description : Sets the mobile's maximum transmitter level.

Parameter : <NR2> Range 15 dBm to 45 dBm, rounded up to the nearest 5dB step.

Allowed suffices : DBM

Example : `:SYSetup:MAxtxlev 23DBM`
Sets the mobile's maximum transmitter level to 25 dBm (rounded up)

:MAxtxlev?

Description : Returns currently set value.

Response : <NR2> Range 15 dBm to 45 dBm in 5dB steps.

Example : `:SYSetup:MAxtxlev?`

Example response : `SYS:MA 25DBM`

:SYSetup

Applicable to GSM, DCS and PCS and TETRA MOBILE

:MCc

Description : Sets the Mobile Country Code.
Parameter : <NR1> Range 0 to 999 for GSM, DCS and PCS
or 1 to 999 for TETRA MOBILE
Allowed suffices : Not applicable
Example : `:SYSetup:MCc 503`
Sets the Mobile Country Code to 503

:MCc?

Description : Returns currently set value.
Response : <NR1> Range 0 to 999 for GSM, DCS and PCS
or 1 to 999 for TETRA MOBILE
Example : `:SYSetup:MCc?`
Example response : `SYS:MC 503`

:SYSetup

Applicable to MPT1327

:MIDent

Description : Sets an MPT1327 mobile's identity.
Parameter : <NR1> Range 0 to 8191
Allowed suffices : Not applicable
Example : `:SYSetup:MIDent 100`
Sets the mobile's identity to 100

:MIDent?

Description : Returns currently set value.
Response : <NR1> Range 0 to 8191
Example : `:SYSetup:MIDent?`
Example response : `SYS:MID 100`

:SYSetup

Applicable to AMPS, TACS and NMT

:MIN

Description : Sets the Mobile Identity Number.

Parameter : <CPD> **Note:** format is system-specific, as shown below:-
1. AMPS, JTACS or NTACS = "*ddd-ddd-dddd*"
2. TACS = "*ddd-d-dddddd*"
3. NMT = "*hhhhhhh*"
where *d* = decimal character
and *h* = hexadecimal character

Allowed suffices : Not applicable

Example : `:SYSetup:MIN "123-456-7890"`
Sets the MIN to 123-456-7890

:MIN?

Description : Returns currently set value.

Response : <CRD> Note: format is system-specific, as shown below.
1. AMPS, JTACS or NTACS = "*ddd-ddd-dddd*"
2. TACS = "*ddd-d-dddddd*"
3. NMT = "*hhhhhhh*"
where *d* = decimal character
and *h* = hexadecimal character.

Example : `:SYSetup:MIN?`

Example response : `SYS:MIN "123-456-7890"`

:SYSetup

Applicable to GSM, DCS, PCS and TETRA MOBILE

:MINRxlev

Description: Sets the mobile's minimum receiver level for access.

Parameter: <NR2> Range –110 dBm to –47 dBm for GSM, DCS and PCS
or –125dBm to –50dBm, rounded up to the nearest 5dB step for TETRA MOBILE

Allowed suffices: DBM

Example: `:SYSetup:MINRxlev -109DBM`
Sets the mobile's minimum receiver level to –108 dBm for GSM, DCS and PCS or –105dBm (rounded up) for TETRA MOBILE)

:MINRxlev?

Description: Returns currently set value.

Response: <NR2> Range –110 dBm to –47 dBm for GSM, DCS and PCS
or –125dBm to –50dBm in 5dB steps for TETRA MOBILE

Example: `:SYSetup:MINRxlev?`

Example response: `SYS:MINR -108DBM`

:SYSetup

Applicable to GSM, DCS, PCS and TETRA MOBILE

:MNc

Description: Sets the Mobile Network Code.

Parameter: <NR1> Range 0 to 99 for GSM, DCS and PCS
or 0 to 16383 for TETRA MOBILE

Allowed suffices: Not applicable

Example: `:SYSetup:MNc 15`
Sets the MNC to 15

:MNc?

Description: Returns currently set value.

Response: <NR1> Range 0 to 99 for GSM, DCS and PCS
or 0 to 16383 for TETRA MOBILE

Example: `:SYSetup:MNc?`

Example response: `SYS:MN 15`

:SYSetup

Applicable to NMT

:MObiletype

Description : Sets the NMT mobile type.
Parameter : <CPD> HMSPORT or ORDINARY
Allowed suffices : Not applicable
Example : `:SYSetup:MObiletype ORDINARY`
Sets the NMT mobile type to be Ordinary

:MObiletype?

Description : Returns currently set value.
Response : <CRD> HMSPORT or ORDINARY
Example : `:SYSetup:MObiletype?`
Example response : `SYS:MO ORDINARY`

:SYSetup

Applicable to MPT1327

:MPrefix

Description : Sets an MPT1327 mobile's prefix.
Parameter : <NR1> Range 0 to 127
Allowed suffices : Not applicable
Example : `:SYSetup:MPrefix 100`
Sets the mobile's prefix to 100

:MPrefix?

Description : Returns currently set value.
Response : <NR1> Range 0 to 127
Example : `:SYSetup:MPrefix?`
Example response : `SYS:MP 100`

:SYSetup

Applicable to GSM, DCS and PCS

:NCc

Description : Sets the Network Colour Code.

Parameter : <NR1> Range 0 to 7

Allowed suffices : Not applicable

Example : `:SYSetup:NCc 50`
Sets the Network Colour Code to 50

:NCc?

Description : Returns currently set value.

Response : <NR1> Range 0 to 7

Example : `:SYSetup:NCc?`

Example response : `SYS:NC 50`

:SYSetup

Applicable to AMPS, NAMPS, MPT1327, TACS and NTACS

:NFormat

Description : Sets the base by which the mobile's ESN is displayed.

Parameter : <CPD> DECIMAL
HEX
OCTAL (AMPS and TACS only)
STANDARD1
STANDARD2 (AMPS and TACS only)

Allowed suffices : Not applicable

Example : `:SYSetup:NFormat HEX`
The received mobile's ESN will now be displayed in Hex

:NFormat?

Description : Returns currently set value.

Response : <CRD> DECIMAL
HEX
OCTAL (AMPS and TACS only)
STANDARD1
STANDARD2 (AMPS and TACS only)

Example : `:SYSetup:NFormat?`

Example response : `SYS:NF HEX`

:SYSetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:PAusemode

Description : Sets the pause condition for Autorun testing.

Parameter : <CPD> MANUAL_ONLY
ON_FAILURE
ALWAYS

Allowed suffices : Not applicable

Example : `:SYSetup:PAusemode ON_FAILURE`
Autorun will now pause when it detects a test failure

:PAusemode?

Description : Returns currently set value.

Response : <CRD> MANUAL_ONLY
ON_FAILURE
ALWAYS

Example : `:SYSetup:PAusemode?`

Example response : `SYS:PA ON_FAILURE`

:SYSetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS and PCS

:PRint

Description : Sets the print condition for Autorun testing.

Parameter : <CPD> OFF, ON

Allowed suffices : Not applicable

Example : `:SYSetup:PRint ON`
Autorun will now print each test result on an external printer

:PRint?

Description : Returns currently set value.

Response : <CRD > OFF, ON

Example : `:SYSetup:PRint?`

Example response : `SYS:PR ON`

:SYSetup

Applicable to GSM, DCS and PCS

:RAND1

Description : Sets Authentication Random Number (part 1).
Parameter : <NR1> Range 0 to #HFFFFFFFF
Allowed suffices : Not applicable
Example : `:SYSetup:RAND1 #H3D4C1EF6`
Sets the Authentication Random Number (part 1) to #H3D4C1EF6

:RAND1?

Description : Returns currently set value.
Response : <NR1> Range 0 to #HFFFFFFFF
Example : `:SYSetup:RAND1?`
Example response : `SYS:RAND1 #H3D4C1EF6`

:SYSetup

Applicable to GSM, DCS and PCS

:RAND2

Description : Sets Authentication Random Number (part 2).
Parameter : <NR1> Range 0 to #HFFFFFFFF
Allowed suffices : Not applicable
Example : `:SYSetup:RAND2 #H3D4C1EF6`
Sets the Authentication Random Number (part 2) to #H3D4C1EF6

:RAND2?

Description : Returns currently set value.
Response : <NR1> Range 0 to #HFFFFFFFF
Example : `:SYSetup:RAND2?`
Example response : `SYS:RAND2 #H3D4C1EF6`

:SYSetup

Applicable to GSM, DCS and PCS

:RAND3

Description : Sets Authentication Random Number (part 3).
Parameter : <NR1> Range 0 to #HFFFFFFFF
Allowed suffices : Not applicable
Example : `:SYSetup:RAND3 #H3D4C1EF6`
Sets the Authentication Random Number (part 3) to #H3D4C1EF6

:RAND3?

Description : Returns currently set value.
Response : <NR1> Range 0 to #HFFFFFFFF
Example : `:SYSetup:RAND3?`
Example response : `SYS:RAND3 #H3D4C1EF6`

:SYSetup

Applicable to GSM, DCS and PCS

:RAND4

Description : Sets Authentication Random Number (part 4).
Parameter : <NR1> Range 0 to #HFFFFFFFF
Allowed suffices : Not applicable
Example : `:SYSetup:RAND4 #H3D4C1EF6`
Sets the Authentication Random Number (part 4) to #H3D4C1EF6

:RAND4?

Description : Returns currently set value.
Response : <NR1> Range 0 to #HFFFFFFFF
Example : `:SYSetup:RAND4?`
Example response : `SYS:RAND4 #H3D4C1EF6`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:REGID

Description : Sets the Registration ID on the test set.

Parameter : <NR1> Range 0 to 1048575 depending on selected system

Allowed suffices : Not applicable

Example : `:SYSetup:REGID 4000`
Sets the Registration ID to 4000

:REGID?

Description : Returns currently set value.

Response : <NR1> Range 0 to 1048575

Example : `:SYSetup:REGID?`

Example response : `SYS:REGID 4000`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:REGINc

Description : Sets the Registration ID increment.

Parameter : <NR1> Range 0 to 4095 depending on selected system

Allowed suffices : Not applicable

Example : `:SYSetup:REGINc 4000`
Sets the Registration ID increment to 4000

:REGINc?

Description : Returns currently set value.

Response : <NR1> Range 0 to 4095

Example : `:SYSetup:REGINc?`

Example response : `SYS:REGIN 4000`

:SYSetup

Applicable to GSM, DCS and PCS

:RESp

Description : Sets the Authentication Response expected from the mobile.
Parameter : <NR1> Range 0 to #HFFFFFFFF
Allowed suffices : Not applicable
Example : `:SYSetup:RESp #H4E6B811B`
Sets the expected Authentication Response to #H4E6B811B

:RESp?

Description : Returns currently set value.
Response : <NR1> Range 0 to #HFFFFFFFF
Example : `:SYSetup:RESp?`
Example response : `SYS:RES #H4E6B811B`

:SYSetup

Applicable to NMT

:SAtphi

Description : Sets the SAT number Φ for the NMT system.
Parameter : <NR1> Range 0 to 4
Allowed suffices : Not applicable
Example : `:SYSetup:SAtphi 2`
Sets the NMT system SAT to 2

:SAtphi?

Description : Returns currently set value.
Response : <NR1> Range 0 to 4
Example : `:SYSetup:SAtphi?`
Example response : `SYS:SA 2`

:SYSetup

Applicable to NMT

:SCAnband

Description : Sets the NMT900 scan band channel type.

Parameter : <CPD> BASIC
CCTC
CC
TC

Allowed suffices : Not applicable

Example : `:SYSetup:SCAnband BASIC`
Sets the NMT scan band channel type to a BASIC channel

:SCAnband?

Description : Returns currently set value.

Response : <CRD> BASIC
CCTC
CC
TC

Example : `:SYSetup:SCAnband?`

Example response : `SYS:SCA BASIC`

:SYSetup

Applicable to AMPS and TACS

:SCC

Description : Sets the system's SAT Colour Code.

Parameter : <NR1> Range 0 to 2

Allowed suffices : Not applicable

Example : `:SYSetup:SCC 2`
Sets the SAT Colour Code to 2

:SCC?

Description : Returns currently set value.

Response : <NR1> Range 0 to 2

Example : `:SYSetup:SCC?`

Example response : `SYS:SCC 2`

:SYSetup

Applicable to AMPS and MPT1327

:SID

Description : Sets the system identity.
Parameter : <NR1> Range 0 to 32767
Allowed suffices : Not applicable
Example : `:SYSetup:SID 3952`
Sets the system identity to 3952

:SID?

Description : Returns currently set value.
Response : <NR1> Range 0 to 32767
Example : `:SYSetup:SID?`
Example response : `SYS:SID 3952`

:SYSetup

Applicable to NMT

:SIGnallingscheme

Description : Sets the user-defined signalling scheme.
Parameter : <CPD> NMT450
NMT450I
Allowed suffices : Not applicable
Example : `:SYSetup:SIGnallingscheme NMT450`
Sets the signalling scheme to NMT450

:SIGnallingscheme?

Description : Returns currently set value.
Response : <CRD> NMT450
NMT450I
Example : `:SYSetup:SIGnallingscheme?`
Example response : `SYS:SIG NMT450`

:SYSetup

Applicable to NMT

:SISc

Description : Enables or disables NMT System Challenge mode.

Parameter : <CPD> DISABLE
ENABLE

Allowed suffices : Not applicable

Example : `:SYSetup:SISc ENABLE`
Enables NMT System Challenge mode

:SISc?

Description : Returns currently set value.

Response : <CRD> DISABLE
ENABLE

Example : `:SYSetup:SISc?`

Example response : `SYS:SIS ENABLE`

:SYSetup

Applicable to NAMPS

:SPWD1

:Ep

Description : Writes to Extended Protocol bit on system parameter word 1.

Parameter : <NR1> Range 0 to 1

Allowed suffices : Not applicable

Example : `:SYSetup:SPWD1:Ep 1`
Sets bit to be a 1

:SPWD1

:Ep?

Description : Returns currently set value.

Response : <NR1> Range 0 to 1

Example : `:SYSetup:SPWD1:Ep?`

Example response : `SYS:SPWD1:E 1`

:SYSetup

Applicable to TACS

:SPWD1

:Pins

Description : Writes authentication bit in system parameter word 1.
Parameter : <NR1> Range 0 to 1
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD1:Pins 1`
Sets bit to be a 1

:SPWD1

:Pins?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1
Example : `:SYSetup:SPWD1:Pins?`
Example response : `SYS:SPWD1:P 1`

:SYSetup

Applicable to TACS2

:SPWD1

:REllevel

Description : Writes to TACS release level bit in system parameter word 1.
Parameter : <NR1> Range 0 to 1
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD1:REllevel 1`
Sets bit to be a 1

:SPWD1

:REllevel?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1
Example : `:SYSetup:SPWD1:REllevel?`
Example response : `SYS:SPWD1:RE 1`

:SYSetup

Applicable to NAMPS and TACS

:SPWD1

:RSVD2

Description : Writes to 2-bit reserved field in system parameter word 1.
Parameter : <NR1> Range 0 to 1
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD1:RSVD2 1`
Sets bit to be a 1

:SPWD1

:RSVD2?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1
Example : `:SYSetup:SPWD1:RSVD2?`
Example response : `SYS:SPWD1:RSVD2 1`

:SYSetup

Applicable to AMPS, NTACS and JTACS

:SPWD1

:RSVD3

Description : Writes to 3-bit reserved field in system parameter word 1.
Parameter : <NR1> Range 0 to 7
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD1:RSVD3 2`
Sets field to be a 2

:SPWD1

:RSVD3?

Description : Returns currently set value.
Response : <NR1> Range 0 to 7
Example : `:SYSetup:SPWD1:RSVD3?`
Example response : `SYS:SPWD1:RSVD3 2`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:CMax

Description : Writes to CMAX-1 field (number of access channels) in system parameter word 2.

Parameter : <NR1> Range 0 to 127

Allowed suffices : Not applicable

Example : `:SYSetup:SPWD2:CMax 10`
Sets the access channels field to 10

:SPWD2

:CMax?

Description : Returns currently set value.

Response : <NR1> Range 0 to 127

Example : `:SYSetup:SPWD2:CMax?`

Example response : `SYS:SPWD2:CM 10`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:CPa

Description : Writes to combined paging/access field in system parameter word 2.

Parameter : <NR1> Range 0 to 1

Allowed suffices : Not applicable

Example : `:SYSetup:SPWD2:CPa 1`
Sets bit to be a 1

:SPWD2

:CPa?

Description : Returns currently set value.

Response : <NR1> Range 0 to 1

Example : `:SYSetup:SPWD2:CPa?`

Example response : `SYS:SPWD2:CP 1`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:Dtx

Description : Writes to discontinuous transmission field in system parameter word 2.
Parameter : <NR1> Range 0 to 1 (0 to 3 for NAMPS)
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD2:Dtx 1`
Sets bit to be a 1

:SPWD2

:Dtx?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1 (0 to 3 for NAMPS)
Example : `:SYSetup:SPWD2:Dtx?`
Example response : `SYS:SPWD2:D 1`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:E

Description : Writes to extended address bit in system parameter word 2.
Parameter : <NR1> Range 0 to 1
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD2:E 1`
Sets bit to be a 1

:SPWD2

:E?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1
Example : `:SYSetup:SPWD2:E?`
Example response : `SYS:SPWD2:E 1`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:Freg

Description : Writes to forced registration bit in system parameter word 2.

Parameter : <NR1> Range 0 to 1

Allowed suffices : Not applicable

Example : `:SYSetup:SPWD2:Freg 1`
Sets bit to be a 1

:SPWD2

:Freg?

Description : Returns currently set value.

Response : <NR1> Range 0 to 1

Example : `:SYSetup:SPWD2:Freg?`

Example response : `SYS:SPWD2:F 1`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:Npaging

Description : Writes to number of paging channels field (N-1) in system parameter word 2.

Parameter : <NR1> Range 0 to 31

Allowed suffices : Not applicable

Example : `:SYSetup:SPWD2:Npaging 10`
Sets the number of paging channels field to 10

:SPWD2

:Npaging?

Description : Returns currently set value.

Response : <NR1> Range 0 to 31

Example : `:SYSetup:SPWD2:Npaging?`

Example response : `SYS:SPWD2:N 10`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:RCf

Description : Writes to read control filler bit in system parameter word 2.
Parameter : <NR1> Range 0 to 1
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD2:RCf 1`
Sets bit to be a 1

:SPWD2

:RCf?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1
Example : `:SYSetup:SPWD1:RCf?`
Example response : `SYS:SPWD2:RC 1`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:REGH

Description : Writes to home registration bit in system parameter word 2.
Parameter : <NR1> Range 0 to 1
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD2:REGH 1`
Sets bit to be a 1

:SPWD2

:REGH?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1
Example : `:SYSetup:SPWD2:REGH?`
Example response : `SYS:SPWD2:REGH 1`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:REGR

Description : Writes to roam registration bit in system parameter word 2.
Parameter : <NR1> Range 0 to 1
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD2:REGR 1`
Sets bit to be a 1

:SPWD2

:REGR?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1
Example : `:SYSetup:SPWD2:REGR?`
Example response : `SYS:SPWD2:REGR 1`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:RSvd

Description : Writes to reserved bit in system parameter word 2.
Parameter : <NR1> Range 0 to 1
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD2:RSvd 0`
Sets bit to be a 0

:SPWD2

:RSvd?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1
Example : `:SYSetup:SPWD2:RSvd?`
Example response : `SYS:SPWD2:RS 0`

:SYSetup

Applicable to AMPS, NAMPS, TACS and NTACS

:SPWD2

:S

Description : Writes to serial number bit in system parameter word 2.
Parameter : <NR1> Range 0 to 1
Allowed suffices : Not applicable
Example : `:SYSetup:SPWD2:S 1`
Sets bit to be a 1

:SPWD2

:S?

Description : Returns currently set value.
Response : <NR1> Range 0 to 1
Example : `:SYSetup:SPWD2:S?`
Example response : `SYS:SPWD1:S 1`

:SYSetup

Applicable to NMT

:TA

Description : Sets the Traffic Area for NMT systems.
Parameter : <NR1> Range 0 to FF, Hex
Allowed suffices : Not applicable
Example : `:SYSetup:TA #Ha2`
Sets the Traffic Area to a2

:TA?

Description : Returns currently set value.
Response : <NR1> Range 0 to FF, Hex
Example : `:SYSetup:TA?`
Example response : `SYS:TA #Ha2`

:SYSetup

Applicable to NMT

:TC

Description : Sets the Tariff Class for NMT systems.
Parameter : <NR1> Range 0 to 255
Allowed suffices : Not applicable
Example : `:SYSetup:TC 100`
Sets the Tariff Class to 100

:TC?

Description : Returns currently set value.
Response : <NR1> Range 0 to 255
Example : `:SYSetup:TC?`
Example response : `SYS:TC 100`

:SYSetup

Applicable to GSM, DCS and PCS

:TCHtsc

Description : Sets the TCH Training Sequence Code.
Parameter : <NR1> Range 0 to 7
Allowed suffices : Not applicable
Example : `:SYSetup:TCHtsc 5`
Sets the TCH Training Sequence Code to 5

:TCHtsc?

Description : Returns currently set value.
Response : <NR1> Range 0 to 7
Example : `:SYSetup:TCHtsc?`
Example response : `SYS:TCH 5`

:SYSetup

Applicable to MPT1327

:TCOn

Description : Specifies the Traffic Confirmation Signalling for MPT1327.

Parameter : <CPD> AHOY
DISABLE
PRESSEL

Allowed suffices : Not applicable

Example : `:SYSetup:TCOn AHOY`
Sets all Traffic Channel Confirmations (i.e. Handoff, Placecall etc.) to require an "AHOY" message from the MPT1327 mobile

:TCOn?

Description : Returns currently set value.

Response : <CRD> AHOY
DISABLE
PRESSEL

Example : `:SYSetup:TCOn?`

Example response : `SYS:TCO AHOY`

:SYSetup

Applicable to all Systems

:TYpe

Description : Sets System Type.

Parameter : <CPD> AMPS, TACS, NMT, PMR, MPT1327, GSM, DCS1800, PCS1900, TETRAMOBILE, TETRABASE, TETRADIRECT or NONE

Allowed suffices : Not applicable

Example : `:SYSetup:TYpe AMPS`
Sets current system to AMPS

:TYpe?

Description : Returns currently set value.

Response : <CRD> AMPS, TACS, NMT, PMR, MPT1327, GSM, DCS1800, PCS1900, TETRAMOBILE, TETRABASE, TETRADIRECT or NONE

Example : `:SYSetup:TYpe?`

Example response : `SYS:TY AMPS`

:SYSetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS, PCS, TETRA MOBILE, TETRA BASE and TETRA DIRECT

:VAriant

Description : Sets System variant.

Parameter : <CPD>

EAMPS, NAMPS, USERAMPS

ETACS, TACS2, CTACS1, CTACS2, JTACS, NTACS USERTACS

NMT450, NMT900, BENELUX, NMTF, AUSTRIA, SPAIN, MALAYSIA, SAUDI_B1, SAUDI_B2, INDONESIA, THAILAND, OMAN, TUNISIA, TURKEY, HUNGARY, RUSSIA, CZECH, BULGERIA, SLOVENIA, USERNMT

BAND3, JRC, HONGKONG, AUTONET, MADEIRA, AMT, NL_TRAXYS, NZ_MPT1327, UK_WATER, PH_INDO, REPART, CARRIS, RADIOMOVEL, USERMPT

P_GSM900, E_GSM900, USERGSM

DCS1800, USERDCS

PCS1900, USERPCS

TETRA_MS, USERTETRA_MS

TETRA_BS, USERTETRA_BS

TETRA_DM, USERTETRA_DM

Allowed suffices : Not applicable

Example : `:SYSetup:VAriant NAMPS`
Sets current System variant to NAMPS

:SYSetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS, PCS, TETRA MOBILE, TETRA BASE and TETRA DIRECT

:VAriant?

Description : Returns currently set value.

Response : <CRD>

EAMPS, NAMPS, USERAMPS

ETACS, TACS2, CTACS1, CTACS2, JTACS, NTACS USERTACS

NMT450, NMT900, BENELUX, NMTF, AUSTRIA, SPAIN, MALAYSIA, SAUDI_B1, SAUDI_B2, INDONESIA, THAILAND, OMAN, TUNISIA, TURKEY, HUNGARY, RUSSIA, CZECH, BULGERIA, SLOVENIA, USERNMT

BAND3, JRC, HONGKONG, AUTONET, MADEIRA, AMT, NL_TRAXYS, NZ_MPT1327, UK_WATER, PH_INDO, USERMPT, REPART, CARRIS, RADIOMOVEL

P_GSM900, E_GSM900, USERGSM

DCS1800, USERDCS

PCS1900, USERPCS

TETRA_MS, USERTETRA_MS

TETRA_BS, USERTETRA_BS

TETRA_DM, USERTETRA_DM

Example : `:SYSetup:VAriant?`

Example response : `SYS:VA NAMPS`

:SYSetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, PMR, GSM, DCS, PCS, TETRA MOBILE, TETRA BASE and TETRA DIRECT

:VErbosemode

Description : Enables or disables the ability to switch to the appropriate menu when a particular parameter is written using a remote command.

Parameter : <CPD> ON, OFF

Allowed suffices : Not applicable

Example : `:SYSetup:VErbosemode ON`
Sets this feature to ON

:VErbosemode?

Description : Returns currently set value.

Response : <CRD> ON, OFF

Example : `:SYSetup:VErbosemode?`

Example response : `SYS:VE ON`

Setup commands for user-defined systems (:SYUsersetup)

List of commands

:SYUsersetup
- :Blockn
  - :CHANNELOffset
  - :CHANNELOffset?
  - :CHANNELSpacing
  - :CHANNELSpacing?
  - :Duplexoffset
  - :Duplexoffset?
  - :Highestchannel
  - :Highestchannel?
  - :Lowestchannel
  - :Lowestchannel?
  - :State
  - :State?
  - :Txbasefreq
  - :Txbasefreq?
- :CONtrolpower
- :CONtrolpower?
- :COUntrycode
- :COUntrycode?
- :Datadevn
- :Datadevn?
- :SAtdevn
- :SAtdevn?
- :Signallingscheme
- :SYNC
- :SYNC?
- :SYNT
- :SYNT?
- :TA
- :TA?
- :TItle
- :TItle?
- :TRafficpower
- :TRafficpower?

:SYUsersetup

Applicable to MPTI327

:Blockn

:CHANNELOffset

Description : For a given user-defined channel block *n*, specifies the offset between logical and physical channel numbers.

Parameters : <NR1> *n* = 1 to 8
<NR1> Range 0 to 9999 (Decimal)

Allowed suffices : Not applicable

Example : `:SYUsersetup:Blockn:CHANNELOffset 1,100`
Sets the logical-to-physical channel offset to be 100 for channel block 1

:Blockn

:CHANNELOffset?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999 (Decimal)

Example : `:SYUsersetup:Blockn:CHANNELOffset? 1`

Example response : `SYU:B:CHANNELO 100`

:SYUsersetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS, PCS, TETRA MOBILE, TETRA BASE and TETRA DIRECT

:Blockn

:CHANNELSpacing

Description : For a given user-defined channel block *n*, specifies the spacing between RF channels.

Parameters : <NR1> *n* = 1 to 8
<NRf> Range ±5 kHz to ±99.999 kHz (except GSM, DCS and PCS)
Range ±5 kHz to ±500 kHz (GSM, DCS and PCS)

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:SYUsersetup:Blockn:CHANNELSpacing 1,30kHz`
Sets the channel spacing for user-defined channel block 1 to 30 kHz

:Blockn

:CHANNELSpacing?

Description : Returns currently set value.

Response : <NR2> Range ±5 kHz to ±99.999 kHz (except GSM, DCS and PCS)
Range ±5 kHz to ±500 kHz (GSM, DCS and PCS)

Example : `:SYUsersetup:Blockn:CHANNELSpacing? 1`

Example response : `SYU:B:CHANNELS 30000.0HZ`

:SYUsersetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS, PCS TETRA MOBILE and TETRA BASE

:Blockn

:Duplexoffset

Description : For a given user-defined channel block *n*, specifies the spacing between forward and reverse channel links for a duplex system.

Parameters : <NR1> *n* = 1 to 8
<NRf> Range –100 MHz to +100 MHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:SYUsersetup:Blockn:Duplexoffset 1,45MHz`
Sets the duplex offset to 45 MHz for channel block 1

:Blockn

:Duplexoffset?

Description : Returns currently set value.

Response : <NR2> Range –100 MHz to +100 MHz

Example : `:SYUsersetup:Blockn:Duplexoffset? 1`

Example response : `SYU:B:D 45000000.0HZ`

:SYUsersetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS, PCS, TETRA MOBILE, TETRA BASE and TETRA DIRECT

:Blockn

:Highestchannel

Description : For a given user-defined channel block *n*, specifies the highest channel for the current channel block number.

Parameters : <NR1> *n* = 1 to 8
<NR1> Range 0 to 9999

Allowed suffices : Not applicable

Example : `:SYUsersetup:Blockn:Highestchannel 1,1000`
Sets the highest channel number to be 1000 for channel block 1

:Blockn

:Highestchannel?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999

Example : `:SYUsersetup:Blockn:Highestchannel? 1`

Example response : `SYU:B:H 1000`

:SYUsersetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS, PCS TETRA MOBILE, TETRA BASE and TETRA DIRECT

:Blockn

:Lowestchannel

Description : For a given user-defined channel block *n*, specifies the lowest channel.

Parameters : <NR1> *n* = 1 to 8
<NR1> Range 0 to 9999

Allowed suffices : Not applicable

Example : `:SYUsersetup:Blockn:Lowestchannel 1,10`
Sets the lowest channel number to 10 for channel block 1

:Blockn

:Lowestchannel?

Description : Returns currently set value.

Response : <NR1> Range 0 to 9999

Example : `:SYUsersetup:Blockn:Lowestchannel? 1`

Example response : `SYU:B:L 10`

:SYUsersetup

Applicable to AMPS, NAMPS, TACS, NTACS,, NMT, GSM, DCS, PCS TETRA MOBILE, TETRA BASE and TETRA DIRECT

:Blockn

:State

Description : For a given user-defined channel block *n*, enables or disables the block.

Parameters : <NR1> *n* = 1 to 8
<CPD> INCLUDED or EXCLUDED

Allowed suffices : Not applicable

Example : `:SYUsersetup:Blockn:State 1,INCLUDED`
Enables channel block 1

:Blockn

:State?

Description : Returns currently set value.

Response : <CRD> INCLUDED or EXCLUDED

Example : `:SYUsersetup:Blockn:State? 1`

Example response : `SYU:B:S INCLUDED`

:SYUsersetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS, PCS TETRA MOBILE, TETRA BASE and TETRA DIRECT

:Blockn

:Txbasefreq

Description : For a given user-defined channel block *n*, specifies the transmitter base frequency for the lowest channel.

Parameters : <NR1> *n* = 1 to 8
<NRf> Range 10.0 MHz to 999.9999 MHz (except DCS/PCS)
Range 10.0 MHz to 2000.0000 MHz (for DCS/PCS)

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:SYUsersetup:Blockn:Txbasefreq 1,45MHz`
Sets the transmitter base frequency for channel block 1 to 45 MHz

:Blockn

:Txbasefreq?

Description : Returns currently set value.

Response : <NR2> Range 10.0 MHz to 999.9999 MHz (except DCS/PCS)
Range 10.0 MHz to 2000.0000 MHz (for DCS/PCS)

Example : `:SYUsersetup:Blockn:Txbasefreq? 1`

Example response : `SYU:B:T 45000000.0HZ`

:SYUsersetup

Applicable to NMT

:CONtrolpower

Description : Sets the reverse Control Channel Power for an NMT mobile.

Parameter : <NR1> Range 0 to 3

Allowed suffices : Not applicable

Example : `:SYUsersetup:CONtrolpower 3`
Sets the reverse Control Channel Power to 3

:CONtrolpower?

Description : Returns currently set value.

Response : <NR1> Range 0 to 3

Example : `:SYUsersetup:CONtrolpower?`

Example response : `SYU:CON 3`

:SYUsersetup

Applicable to NMT

:COUntrycode

Description : Sets the operational Country Code for the user-defined NMT system.

Parameter : <NR1> Range 0 to 15

Allowed suffices : Not applicable

Example : `:SYUsersetup:COUntrycode 15`
Sets the Country Code to 15

:COUntrycode?

Description : Returns currently set value.

Response : <NR1> Range 0 to 15

Example : `:SYUsersetup:COuntrycode?`

Example response : `SYU:COU 15`

:SYUsersetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327 and NMT

:Datadevn

Description : Sets the Data deviation level for both control and voice (traffic) channels.

Parameter : <NRf> Range 0 to 8 kHz

Allowed suffices : `HZ, KHZ, MHZ`

Example : `:SYUsersetup:Datadevn 5kHz`
Sets the Data deviation level to 5 kHz

:Datadevn?

Description : Returns currently set value.

Response : <NR2> Range 0 to 8 kHz

Example : `:SYUsersetup:Datadevn?`

Example response : `SYU:D 5000.0HZ`

:SYUsersetup

Applicable to AMPS, NAMPS, TACS, NTACS and NMT

:SAtdevn

Description : Sets the SAT deviation level for voice traffic channels.
Parameter : <NRf> Range 0 to 2.5 kHz
Allowed suffices : `HZ, KHZ, MHZ`
Example : `:SYUsersetup:SAtdevn 1.7kHz`
Sets the SAT deviation level to 1.7 kHz

:SAtdevn?

Description : Returns currently set value.
Response : <NR2> Range 0 to 2.5 kHz
Example : `:SYUsersetup:SAtdevn?`
Example response : `SYU:SA 1700.0HZ`

:SYUsersetup

Applicable to NMT

:SIgnallingscheme

Description : Sets the user-defined signalling scheme.
Parameter : <CPD> NMT450, NMT450I and NMT900
Allowed suffices : Not applicable
Example : `:SYUsersetup:SIgnallingscheme NMT450`
Sets the signalling scheme to NMT450

:SIgnallingscheme

Description : Returns currently set value.
Response : <CRD> NMT450, NMT450I and NMT900
Example : `:SYUserSetup:SIgnallingscheme?`
Example response : `SYU:SI NMT450`

:SYUsersetup

Applicable to MPT1327

:SYNC

Description : Sets the user-defined MPT1327 Control Channel Sync pattern.
Parameter : <NR1> Range 0 to FFFF, Hex
Allowed suffices : Not applicable
Example : `:SYUsersetup:SYNC #Hcd17`
Sets the Control Channel Sync pattern to CD17

:SYNC?

Description : Returns currently set value.
Response : <NR1> Range 0 to FFFF, Hex
Example : `:SYUsersetup:SYNC?`
Example response : `SYU:SYNC #Hcd17`

:SYUsersetup

Applicable to MPT1327

:SYNT

Description : Sets the user-defined MPT1327 Traffic Channel Sync pattern.
Parameter : <NR1> Range 0 to FFFF, Hex
Allowed suffices : Not applicable
Example : `:SYUsersetup:SYNT #Hcd17`
Sets the Traffic Channel Sync pattern to CD17

:SYNT?

Description : Returns currently set value.
Response : <NR1> Range 0 to FFFF, Hex
Example : `:SYUsersetup:SYNT?`
Example response : `SYU:SYNT #Hcd17`

:SYUsersetup

Applicable to NMT

:TA

Description : Sets the user-defined NMT Y1 Traffic Area.
Parameter : <NR1> Range 0 to F, Hex
Allowed suffices : Not applicable
Example : `:SYUsersetup:TA #H3`
Sets Y1 to 3

:TA?

Description : Returns currently set value.
Response : <NR1> Range 0 to F, Hex
Example : `:SYUsersetup:TA?`
Example response : `SYU:TA #H3`

:SYUsersetup

Applicable to AMPS, NAMPS, TACS, NTACS, MPT1327, NMT, GSM, DCS, PCS TETRA MOBILE, TETRA BASE and TETRA DIRECT

:TItle

Description : Sets the User Title for the user-defined system.
Parameter : <CPD> Range 1 to 10 characters
Allowed suffices : Not applicable
Example : `:SYUsersetup:TItle "FRED"`
Sets the User Title to "FRED"

:TItle?

Description : Returns currently set value.
Response : <CRD> Range 1 to 10 characters
Example : `:SYUsersetup:TItle?`
Example response : `SYU:TI "FRED"`

:SYUsersetup

Applicable to NMT

:TRafficpower

Description : Sets the reverse Traffic Channel Power for an NMT mobile.

Parameter : <NR1> Range 0 to 3

Allowed suffices : Not applicable

Example : `:SYUsersetup:TRafficpower 3`
Sets the reverse Traffic Channel Power to 3

:TRafficpower?

Description : Returns currently set value.

Response : <NR1> Range 0 to 3

Example : `:SYUsersetup:TRafficpower?`

Example response : `SYU:TR 3`

MP MASTER
MA31138